

RELIABILITY ENHANCEMENT OF NAVIER-STOKES CODES THROUGH CONVERGENCE ACCELERATION

NASA CONTRACT NAS8-38861

Charles L. Merkle/George S. Dulikravich
The Pennsylvania State University
University Park, PA 16802

FINAL REPORT

Submitted to:

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

April, 1995

N95-30034

Unclass

63/34 0054609

(NASA-CR-196635) RELIABILITY
ENHANCEMENT OF NAVIER-STOKES CODES
THROUGH CONVERGENCE ACCELERATION
Final Report (Pennsylvania State
Univ.) 223 p

TABLE OF CONTENTS

	Page
Reliability Enhancement of Navier-Stokes Algorithms Through Convergence Acceleration.....	1
Sensitivity-Based Methods for Convergence of Iterative Algorithms.....	155
Acceleration of Iterative Algorithms on Highly Clustered Grids.....	185

**Reliability Enhancement of Navier-Stokes Algorithms
Through Convergence Acceleration**

Final Report

C. L. Merkle, S. Venkateswaran and P. E. O. Buelow

Table of Contents

1	INTRODUCTION	4
1.1	Sources of Convergence Difficulties and Mitigation Methods	4
1.2	Stability Theory and its Role in Understanding Convergence	7
2	PRECONDITIONED EQUATIONS OF MOTION	13
2.1	Preconditioned System	13
2.2	Implicit-ADI Algorithm	15
2.3	Explicit R-K Algorithm	15
2.4	Stability Results of Preconditioned System	16
3	ASSESSMENT OF PRESSURE-BASED ALGORITHMS	18
3.1	Introduction	18
3.2	Vector Formulation of PISO Algorithm	18
3.2.1	Uncoupling the PISO Algorithm	21
3.2.2	Diagonally Dominant PISO Scheme	22
3.3	PISO Stability Results	23
3.3.1	Conditional Stability of the PISO Scheme	23
3.3.2	PISO At Different Mach Numbers	25
4	VISCOUS PRECONDITIONING	35
4.1	The 1-D Navier Stokes Equations	35
4.2	The Preconditioned Navier-Stokes Equations	37
4.3	Formulation of Viscous Preconditioning	37
4.4	Von Neumann Stability Results	39
4.5	Computational Results	40
5	GRID ASPECT RATIO EFFECTS ON CONVERGENCE OF CENTRAL DIFFERENCED ALGORITHMS	47
5.1	Time-Step Definition and the Problem at High Aspect Ratios	47
5.2	Analysis of Euler Equations	48
5.3	Analysis of the Navier-Stokes Equations	49
5.4	Application of Boundary Conditions	51
5.5	Multi-Stage Explicit Schemes	52
5.6	Computational Results	53
5.6.1	Inviscid Flow in Straight Duct	53
5.6.2	Viscous Flow in Straight Duct	54
5.6.3	Flat Plate Boundary Layer	55
5.6.4	Laminar Backward-Facing Step	56

5.6.5 Converging-Diverging Nozzle

57

6 PRECONDITIONING FOR TURBULENT AND REACTING FLOWS

74

6.1 Governing Equations

74

6.2 Derivation of the Preconditioned System

75

6.3 Computational Results

78

6.3.1 Multi-Species Non-Reacting Shear Layers

79

6.3.2 Reacting Shear Layers

79

6.3.3 Turbulent Shear Layers

80

6.3.4 Turbulent Flow Over a Backward-facing Step

80

7 APPLICATION OF PRECONDITIONING TO UPWIND SCHEMES

95

7.1 Formulation of Upwind Schemes

95

7.1.1 Flux-Difference Splitting Procedure

95

7.1.2 Approximate-Inversion Methods

96

7.2 Results

99

7.2.1 Euler Stability Analysis

99

7.2.1.1 Direct Inversion

100

7.2.1.2 Point Gauss-Seidel (LU) Scheme

101

7.2.1.3 Line Gauss-Seidel (LGS) Scheme

101

7.2.1.4 ADI Algorithm

102

7.2.2 Convergence Results—Euler Equations

102

7.2.3 Navier-Stokes Stability Analysis

103

7.2.3.1 Point Gauss-Seidel Scheme

104

7.2.3.2 Line Gauss-Seidel Scheme

104

7.2.4 Convergence Examples – Navier-Stokes Equations

105

7.2.5 Timing Study

106

8 PRECONDITIONING FOR UNSTEADY COMPUTATIONS

112

8.1 Introduction

112

8.2 Theoretical Development

113

8.2.1 Unsteady Euler Equations

113

8.2.2 Approximate-Newton Method

114

8.2.3 Dual-Time-Stepping Scheme

115

8.2.4 Analysis of the Dual-Time Scheme

118

8.3 Computational Results

120

8.3.1 Acoustic Oscillations in a Cylindrical Chamber

121

8.3.2 Propagation of Lamb Vortex

122

8.3.3 Spatial Instability of Mixing Layer

123

9	GRID ASPECT RATIO EFFECTS ON 3D COMPUTATIONS	133
9.1	Introduction	133
9.2	Three-Factor ADI Scheme	134
9.3	Two Factor ADI Scheme for the 3-D Euler Equations	136
9.4	Formulation of Dual-Time-Stepping Scheme	136
9.5	Computational Results	139
10	CONCLUSIONS	148
	Bibliography	151

INTRODUCTION

1.1 Sources of Convergence Difficulties and Mitigation Methods

Advances in CFD algorithms and enhanced computer capabilities have led to the development of multi-dimensional Navier-Stokes codes which are routinely used for flowfield computations in a wide variety of engineering problems. Furthermore, Navier-Stokes computer codes have been coupled with auxiliary physical equations for turbulence and chemical kinetics and are in wide use. The advent of these advanced Navier-Stokes codes and methods has led to their application to more and more challenging problems and has provided an ever-growing role for CFD in engineering design. Unfortunately, more complex CFD models as well as more complex applications increase the likelihood of code unreliability and require more user intervention during the course of the solution. This trend is, however, exactly opposite to what is required of design tools. They must work efficiently with minimal user intervention. The objectives of the present research are to develop methods for enhancing the reliability of Navier-Stokes codes through understanding and improving convergence characteristics, thereby improving their effectiveness in a design environment.

It is common experience that for simple flowfields with simple geometries standard CFD algorithms generally converge efficiently, are robust and pose no problem to the user. As the complexity of the flowfield increases, however, the robustness and convergence efficiency generally begin to disappear. In fact, even with the convergence slowdown experienced for complex problems, the user intervention required to obtain a final solution can dominate the computational resources and become the most costly part of the solution. There are several contributing phenomena that can lead to this degradation in overall robustness. For instance, code performance may be compromised by the lack of proper grid resolution, the lack of appropriate initial conditions or because all or part of the flowfield lies outside the flow regime in which the algorithm converges efficiently. The last issue is intimately related to the physics of the flowfields under consideration.

Propulsion-related flowfields are governed by a wide variety of physical processes. For instance, in a rocket combustor, flow speeds are low, viscous effects dominate, recirculation regions are often present and combustion results in large rates of energy release and species generation. In contrast, downstream of the combustion zone, in the nozzle, there is strong acceleration of the flow to sonic and supersonic velocities; typically, the Reynolds numbers are high and heat release terms are small. Associated with the various physical processes are specific time scales which describe the rate of evolution of these processes. In general, the time scales of the different physical processes are vastly different. This disparity presents a formidable problem to numerical algorithms as convergence of the flowfield solutions is

typically controlled by the rate of the slowest evolving processes. The problem is commonly referred to as a *stiffness* in the governing equations.

In this research work, we focus on several common sources of equation stiffness that lead to convergence degradation of most CFD algorithms. These include the convergence difficulties due to low speeds, viscous-dominated regions, strongly stretched grid cells, chemical reactions and turbulence. In all cases, we employ von Neumann stability theory as a tool to study the numerical difficulties involved. Even though the stability theory is strictly valid only for linear, constant-coefficient problems, it is extremely useful in understanding numerical convergence and stability behaviour. It should be emphasized that the stability characteristics not only reveal whether a given algorithm is stable or not but also provide insight into numerical convergence behaviour. Using stability results as the basis, we then devise improved algorithms to eliminate the stiffness problems and thereby ensure efficient convergence over a wide range of physical situations. In order to test these improvements, we employ a hierarchy of test cases, starting with simple test problems and going on to more complicated flowfields. The simple problems allow us to carry out parametric studies varying only one parameter at a time. This allows us to establish a close correspondence between stability theory and numerical convergence. The more complex flowfields enable us to confirm that the methods retain their effectiveness in practical situations.

There are two basic algorithmic approaches in CFD— pressure-based and density-based methods. Each of these classes of methods have their inherent strengths and weaknesses. Pressure-based methods were originally developed as iterative schemes for incompressible flows at low Reynolds numbers [1] and were then extended to high Reynolds number flows and compressible applications [2]. On the other hand, density-based methods were originally developed as time-marching schemes for inviscid and high Reynolds number compressible transonic flows [3, 4, 5]. The classic difficulty with density-based methods is the presence of extended low Mach number regions where the disparity between the acoustic and particle speeds typically causes convergence deterioration. Recent research into this problem by several workers including the present authors has largely negated this difficulty through the introduction of time-derivative preconditioning [6, 7, 8, 9, 10, 11]. Preconditioning alters the acoustic speeds of the system, making them the same order of magnitude as the gas velocities. Convergence rates that are independent of Mach number are thereby obtained. In our research, we employ the preconditioned Navier-Stokes equations as the basic computational framework. We carry out studies of stability behaviour and numerical convergence characteristics and, in addition, also use it as a basis for comparing the density-based and the pressure-based methods.

Pressure-based and preconditioned methods are related in philosophy [12]. Pressure-based methods replace the standard continuity equation with a second-order equation for the pressure field. This equation represents the propagation of acoustic waves and has a characteristic speed that represents an equivalent speed of sound. The remaining momentum and energy equations have a characteristic speed given by the fluid velocity. The equations are then solved sequentially. By selecting the appropriate time-step definition (or by using an iterative solution scheme) for each individual equation, optimum convergence may

be obtained at low flow speeds. This procedure is equivalent (in principle) to the explicit alteration of the acoustic speeds by preconditioning techniques. In other words, the preconditioning method becomes essentially ‘pressure-based’ at low Mach numbers but reverts to the traditional density-based form at transonic and supersonic speeds. The similarities and differences between these approaches are discussed in greater detail in a later section.

A time-scale disparity similar to that for low Mach number flows is also observed in viscous-dominated flowfields. This again leads to convergence deterioration of density based methods. At low Reynolds numbers (typically when the grid cell Reynolds number becomes of order one), there is a strong disparity between the diffusion (viscous) time scale and the convective (inviscid) time scale. Such low Reynolds number regions occur even in high Reynolds number flows such as boundary layers, in recirculation zones or in other regions where the flow field gradients are strong. In a later section, we discuss how time-derivative preconditioning may be fashioned such that the acoustic time scale is of the same order as the diffusion time scale. The resulting viscous-preconditioning matrix is shown to be effective down to extremely low Reynolds number situations.

Another factor that can play a crucial role in the convergence of Navier-Stokes computations is the sensitivity to grid-stretching and high grid aspect ratios. This situation is particularly characteristic of turbulent flow calculations where the grid must be refined very tightly in the dimension normal to the wall to resolve the steep velocity gradient. A second type of problem in which high aspect ratio grids are encountered is in flowfields that are very long in relation to their cross-sectional dimensions as, for example, internal flows in ducts. In both instances, the high aspect ratio of the grid cells causes strong disparity in the wave propagation speeds (or, more precisely, in the time scales) in the two coordinate directions causing serious convergence deterioration. The specific reasons for this convergence degradation is analyzed for a wide range of numerical schemes ranging from central-differenced schemes to various upwind-based schemes. Both explicit and implicit schemes are considered. For implicit methods, it is shown that fundamental algorithmic modifications provide effective convergence behaviour for grids from aspect ratio unity to the limit of very high aspect ratio grids.

The preconditioning techniques and algorithms derived for Navier-Stokes computations may be readily extended to more general problems involving additional transport equations. Here, we consider the examples of turbulent and reacting flows, wherein additional differential equations need to be solved for describing the transport of turbulent kinetic energy, dissipation rate and chemical species. A generalized form of the preconditioning matrix is derived for handling such problems and, in all cases, significant improvements are obtained over standard methods. Further, we also consider algorithms for unsteady computations and discuss the special numerical issues that arise here. Again, the preconditioning methodology may be employed to improve computational efficiency of unsteady flow problems as well. Finally, we discuss the extensions of these ideas to three-dimensional flow problems.

1.2 Stability Theory and its Role in Understanding Convergence

Stability analyses of numerical schemes have two distinctly different uses. First of all, they predict whether a numerical algorithm will be stable or unstable, but perhaps more importantly, they also predict the rate of decay of errors, and therefore the convergence rate of the scheme. The von Neumann (or Fourier) stability analysis will be used as a diagnostic tool to examine the stability of various schemes and the potential impact of any modifications.

This analysis is applicable only to linear, constant coefficient equations with periodic boundaries. While the conditions on the von Neumann analysis sound restrictive, the results are actually quite useful. Comparisons with numerical experiments reveal that the von Neumann stability analysis does a good job of predicting relative convergence rates, and whether a scheme will be stiff or even divergent. The results from the von Neumann analysis will often give a worst-case scenario. Since the analysis requires periodic boundaries, the effect of errors propagating out of the computational domain is not taken into account. Numerical damping is the only predicted means for reducing the errors. Most multi-dimensional numerical algorithms employ boundary conditions that permit only part of the errors to exit the domain. Those errors that do not exit are reflected back into the domain, and numerical damping becomes the primary means for reducing these reflected errors. In this case the von Neumann analysis should give good predictions to the actual convergence.

The stability characteristics of a system of equations are determined by the eigenvalues of the amplification matrix that relates the solution at the new time level to that at the old level. If we designate this matrix by G , we have, $Q^{n+1} = GQ^n$. This amplification matrix is readily obtained from the linearized form of the discretized equations of motion. It is generally necessary to evaluate the amplification matrix numerically to ascertain its characteristics. Corresponding to each equation of motion is a stability eigenvalue at every wave number.

Representative stability results for the one-dimensional Euler equations using Euler implicit differencing in time and central differencing in space are shown in Figures 1.1 and 1.2. Figure 1.1 shows all three eigenvalues of the amplification matrix for flow at a Mach number of 0.4 and a CFL_{u+c} of 5. Here, the CFL is based on the largest acoustic eigenvalue of the system ($CFL_{u+c} = \frac{(u+c)\Delta t}{\Delta x}$). At this value of CFL_{u+c} , the CFL's which correspond to the other wave speeds are $CFL_u=1.4$ and $CFL_{u-c}=2.1$. Here, all three eigenvalues are roughly the same order of magnitude and yield good damping.

The amplification factors shown in Figure 1.2(a) are for a Mach number of 0.01 and a CFL_{u+c} of 5 ($CFL_u=0.05$, $CFL_{u-c}=4.9$). At this low value of the Mach number, the eigenvalues now differ by two orders of magnitude. Even though two of the eigenvalues (g_{u+c} and g_{u-c}) display good damping of errors, the third eigenvalue (g_u) is extremely stiff (i.e. $|g_u|$ is very close to unity) and will clearly control the convergence of the solution. This stiffness is related to the very small value of CFL_u .

Since the Euler implicit algorithm applied to the 1-D Euler equations is unconditionally stable, and in this case numerical damping of all wave-numbers improves as the CFL

increases, this stiffness could be alleviated simply by increasing the CFL numbers. Figure 1.2(b) shows the amplification factors for $\text{CFL}_{u+c}=500$ ($\text{CFL}_u=5$, $\text{CFL}_{u-c}=490$), where the stiffness no longer exists. This feature of improved damping with increasing CFL is often lost for higher dimensions, because multi-dimensional algorithms employ approximate factorization or relaxation techniques which introduce errors because they do not provide an exact solution to the implicit operator. These errors are small at low to moderate CFL's, but can become significant at large CFL's causing the scheme to become stiff or even unstable.

Contours of the maximum eigenvalue of the amplification matrix are also shown for the two-dimensional Euler equations in Figures 1.3 and 1.4. In these figures, Euler implicit time-differencing is used along with central-differencing for the spatial derivatives. Direct inversion (i.e. no factorization) of the implicit, left-hand-side (LHS) operator is modeled. Results for two different Mach numbers are presented in each figure by taking advantage of symmetry about the leading diagonal. Figure 1.3 presents the amplification factors for $M=0.3$ (upper triangle) and $M=0.03$ (lower triangle) at a CFL of unity. The amplification factors for the $M=0.3$ case suggest better damping characteristics than the $M=0.03$ case, showing the increase in stiffness as the Mach number decreases. Improved damping can be obtained by simply increasing the CFL, as shown in Figure 1.4. Here, the $\text{CFL}=10$, and the Mach numbers are the same as the previous figure. For the Mach 0.3 case, the minimum amplification factor has decreased to 0.3, while at the Mach 0.03 case, the minimum amplification factor has decreased to 0.91. Larger time steps will eventually offset the detrimental effects of low Mach numbers, but as mentioned earlier, the optimum time step is limited by approximate factorization. Recall that these present results are for the fully implicit solution and do not include any factorization errors.

In the following chapters, we present preconditioning schemes that ensure well-conditioned eigenvalues at all flow speeds and Reynolds numbers so convergence becomes independent of Mach number and Reynolds number.

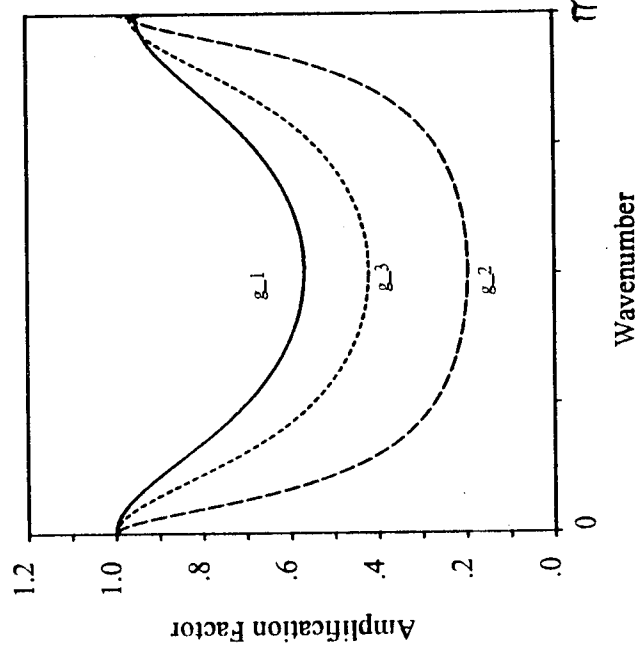


Fig. 1.1. Magnitudes of the eigenvalues of the amplification matrix for the Euler implicit, central-difference scheme applied to the 1-D Euler equations. Mach=0.4, CFL=5.

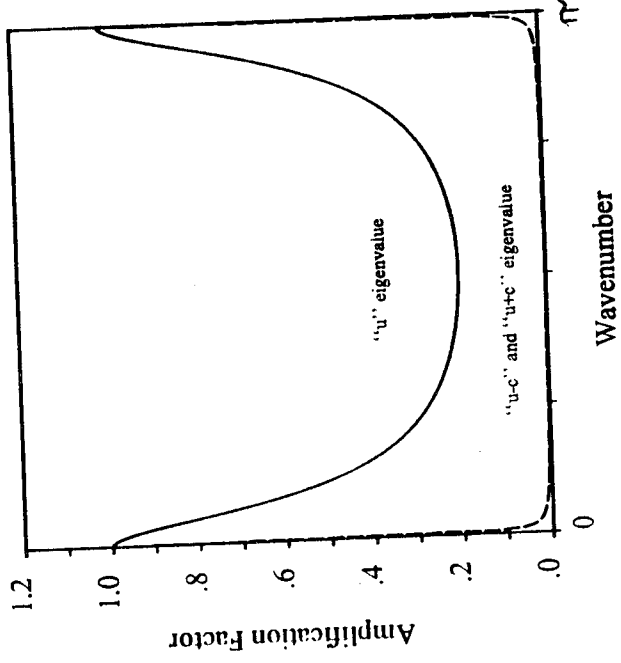
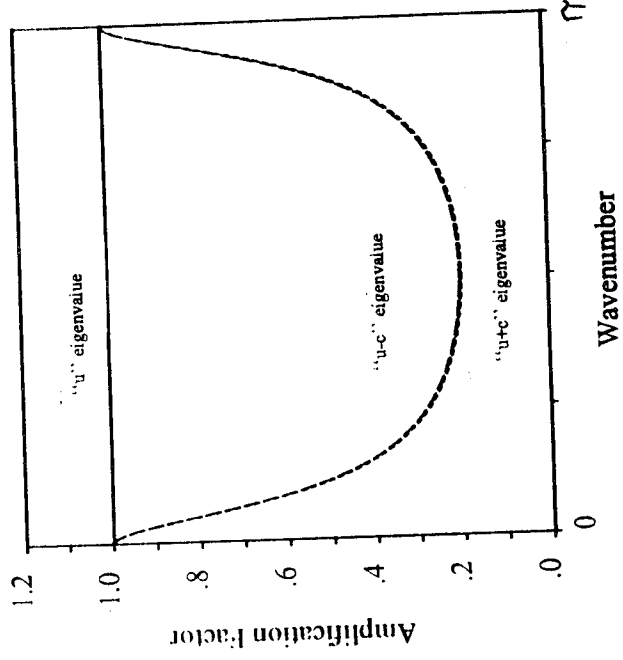


Fig. 1.2. Magnitudes of the eigenvalues of the amplification matrix for the Euler implicit, central-difference scheme applied to the 1-D Euler equations. (a) Mach=0.01, CFL=5, (b) Mach=0.01, CFL=500.

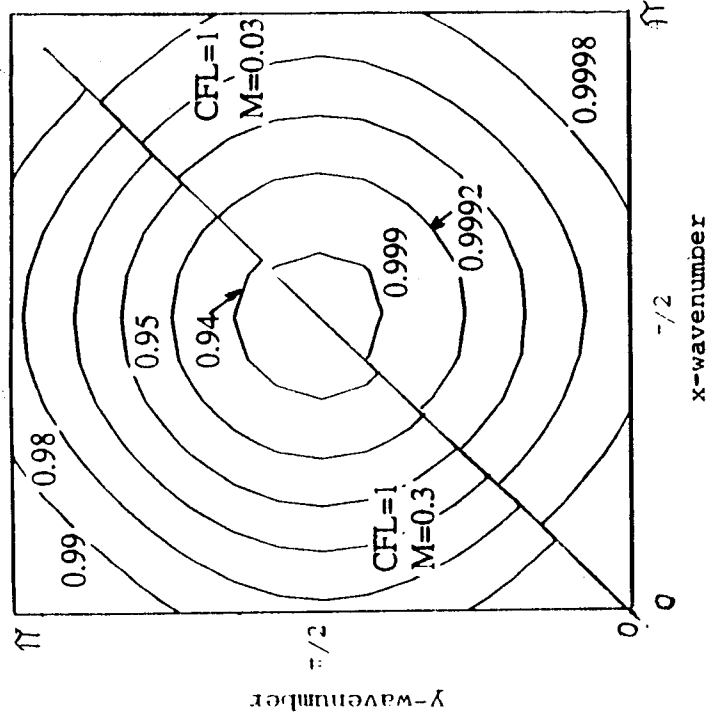


Fig. 1.3. Effect of Mach number on amplification factors of the 2-D Euler implicit, central-difference algorithm, CFL=1. Upper triangle, $M=0.3$; lower triangle, $M=0.03$

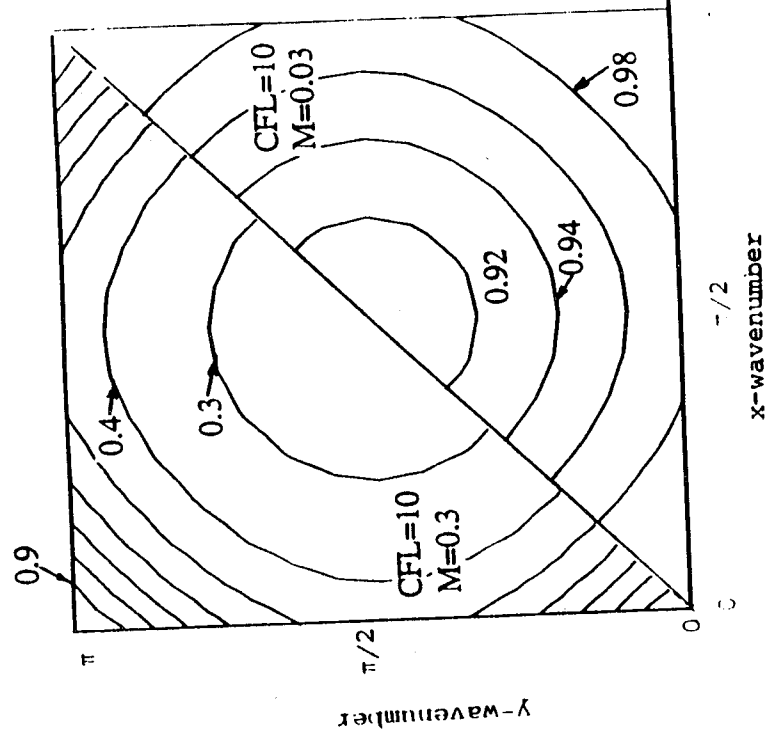


Fig. 1.4. Effect of Mach number on amplification factors of the 2-D Euler implicit, central-difference algorithm, CFL=10. Upper triangle, $M=0.3$; lower triangle, $M=0.03$

PRECONDITIONED EQUATIONS OF MOTION

2.1 Preconditioned System

The Navier-Stokes equations can be written as:

$$\Gamma \frac{\partial Q_v}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = H + \frac{\partial}{\partial x} \left(R_{xx} \frac{\partial}{\partial x} Q_v + R_{xy} \frac{\partial}{\partial y} Q_v \right) + \frac{\partial}{\partial y} \left(R_{yx} \frac{\partial}{\partial x} Q_v + R_{yy} \frac{\partial}{\partial y} Q_v \right) \quad (2.1)$$

where the flux vectors E and F have their traditional definitions for the equations in conservative form,

$$E = \begin{pmatrix} \rho u \\ (\rho u^2 + p) \\ \rho uv \\ (e + p)u \end{pmatrix}, \quad F = \begin{pmatrix} \rho v \\ \rho uv \\ (\rho v^2 + p) \\ (e + p)v \end{pmatrix}$$

and the viscous matrices are given by,

$$\begin{aligned} R_{xx} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{4}{3}\mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & \frac{4}{3}\mu u & \mu v & K \end{pmatrix}, & R_{xy} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{2}{3}\mu \\ 0 & \mu & 0 & 0 \\ 0 & \mu v & -\frac{2}{3}\mu u & 0 \end{pmatrix}, \\ R_{yx} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & -\frac{2}{3}\mu & 0 & 0 \\ 0 & -\frac{2}{3}\mu v & \mu u & 0 \end{pmatrix}, & R_{yy} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \frac{4}{3}\mu & 0 \\ 0 & \mu u & \frac{4}{3}\mu v & K \end{pmatrix}, \end{aligned}$$

For convenience, the traditional primary dependent variable has been replaced by the 'viscous' vector

$$Q_v = (p, u, v, T)^T$$

which occurs naturally inside the diffusion terms, but is selectively placed inside the time-derivative. This choice of primary dependent variable does not impact the unsteady characteristics of the governing equations, but it is advantageous for flows with chemical reactions or arbitrary equations of state. It is also useful in the proper choice of time-derivative preconditioning of the system.

For the standard Navier-Stokes equations, the matrix Γ pre-multiplying the time-derivative is obtained from the chain rule by setting $\Gamma = \Gamma_e = \partial Q / \partial Q_v$, where $Q = (\rho, \rho u, \rho v, e)^T$. This matrix is given as:

$$\Gamma_e = \begin{bmatrix} \frac{1}{RT} & 0 & 0 & -\frac{\rho}{T} \\ \frac{\rho u}{RT} & \rho & 0 & -\frac{\rho u}{T} \\ \frac{\rho v}{RT} & 0 & \rho & -\frac{\rho v}{T} \\ \frac{H}{RT} - 1 & \rho u & \rho v & \rho C_p - \frac{\rho H}{T} \end{bmatrix} \quad (2.2)$$

where H is the total enthalpy per unit mass. The eigenvalues of the inviscid part of Eqn. (2.1) are u , $u \pm c$ is the x-direction and v , $v \pm c$ in the y-direction. These eigenvalues correspond to the acoustic and particle wave speeds in the system. During the numerical time-marching procedure, these waves are responsible for transporting errors through the solution domain. At transonic speeds, the acoustic and particle wave-speeds are well-conditioned and good convergence behaviour is observed. However, at low speeds, the acoustic speeds are much higher than the particle velocities and this disparity is responsible for poor convergence rates. In the latter instance, it becomes necessary to control the conditioning of the wave-speeds in order to control convergence behaviour.

An effective way of controlling the characteristic speeds of the system is to replace the physical matrix Γ with an appropriate artificial matrix, Γ_p , referred to as the preconditioning matrix. One choice of this matrix, which ensures efficient convergence at all Mach numbers, is derived by scaling the time-derivative of pressure which establishes the speed of acoustic waves in the system. This preconditioning matrix is given as:

$$\Gamma_p = \begin{bmatrix} \frac{1}{\epsilon \gamma RT} & 0 & 0 & 0 \\ \frac{\rho u}{\epsilon \gamma RT} & \rho & 0 & 0 \\ \frac{\rho v}{\epsilon \gamma RT} & 0 & \rho & 0 \\ \frac{H}{\epsilon \gamma RT} - 1 & \rho u & \rho v & \rho C_p \end{bmatrix} \quad (2.3)$$

The preconditioning parameter ϵ for inviscid flows is chosen to maintain well-conditioned eigenvalues at all Mach numbers. The eigenvalues of the preconditioned system in the x-direction are given by:

$$\lambda_{1,2} = u \quad \lambda_{3,4} = \frac{1}{2} [u(1 + \epsilon) \pm \sqrt{u^2(1 - \epsilon)^2 + 4\epsilon c^2}] \quad (2.4)$$

The first two eigenvalues represent the speeds of the particle waves, while the third and fourth eigenvalues are the acoustic wave-speeds. Examination of Eqn. (2.4) suggests the following choice for ϵ in order to maintain well-conditioned wave-speeds at all Mach numbers:

$$\epsilon = M_r^2 = \begin{cases} \epsilon^2, & M \leq \epsilon \\ M^2, & \epsilon < M < 1 \\ 1, & M \geq 1 \end{cases} \quad (2.5)$$

For subsonic Mach numbers, the above choice always maintains the acoustic eigenvalues to be of the same order as the particle velocity. For supersonic Mach numbers, $\epsilon = 1$ and the

acoustic eigenvalues revert to their standard values of $u \pm c$. In other words, when $\epsilon = 1$, the preconditioned system is essentially the same as the standard non-preconditioned system, although some differences remain in the form of the time-derivatives.

2.2 Implicit-ADI Algorithm

Steady-state solutions of Eqn. (2.1) are obtained by using Euler implicit temporal discretization and second-order central-differencing for the spatial discretizations. The resulting matrix operator may be approximately factored for efficient inversion by using the Douglas-Gunn Alternating Direction Implicit (ADI) procedure [3, 4, 13]. When expressed in delta form, this becomes:

$$\left[S + \frac{\partial A}{\partial x} - \frac{\partial}{\partial x} R_{xx} \frac{\partial}{\partial x} \right] S^{-1} \left[S + \frac{\partial B}{\partial y} - \frac{\partial}{\partial y} R_{yy} \frac{\partial}{\partial y} \right] \Delta Q_v = -\mathcal{R}^n \quad (2.6)$$

where the residual $\mathcal{R}^n$ is the steady state version of Eqn. (2.1) and the remaining terms are defined as:

$$S = \frac{\Gamma}{\Delta t} - D, \quad A = \frac{\partial E}{\partial Q_v}, \quad B = \frac{\partial F}{\partial Q_v}, \quad D = \frac{\partial H}{\partial Q_v}$$

2.3 Explicit R-K Algorithm

Even though the main focus of this research is implicit techniques, for the sake of completeness, we also present the explicit Runge-Kutta scheme. The four stage Runge Kutta scheme is written as:

$$[\Gamma - \alpha_k \Delta t D] \Delta Q_v = -\alpha_k \Delta t \mathcal{R}^k \quad (2.7)$$

where k is the number of the stage ($k = 1, \dots, 4$),

$$\alpha_1 = 1/4, \quad \alpha_2 = 1/3, \quad \alpha_3 = 1/2, \quad \alpha_4 = 1$$

$$\Delta Q_v = Q_v^k - Q_v^n, \quad Q_v^{n+1} = Q_v^4$$

$\mathcal{R}$ is the residual of Eqn. (2.1) for each intermediate stage. The viscous fluxes, source terms and the source term Jacobian D are evaluated only at the $k = 1$ stage for each time step. Standard second-order and fourth-order dissipation terms are added on the right hand side (also evaluated only at the $k = 1$ stage). The scheme given in Eqn. (2.7) differs from the standard Runge-Kutta algorithm only in the inclusion of the preconditioning matrix Γ and in the Jacobian matrix D .

2.4 Stability Results of Preconditioned System

Stability results are presented for the Euler implicit algorithm with the ADI factorization. Figure 2.1 shows contours of the maximum eigenvalue of the amplification matrix for $M=0.5$, $CFL=5$. Comparing this figure with Figures 1.3 and 1.4 reveals that the ADI factorization errors are affecting the amplification factors in the mid wave-number ($\pi/2$, $\pi/2$) region. In fact if the CFL exceeds a value of 10, the factorization errors will cause the amplification factors to approach unity over most of the wave-number domain, indicating that the algorithm is very stiff.

Figure 2.2 again makes use of symmetry to present two cases. Here the Mach number is 0.01, and the CFL_{u+c} is 5. Results in the upper triangle were generated with the standard equations (i.e. no preconditioning) and show the stiffness which is caused by the low Mach number. The results in the lower triangle were generated using inviscid preconditioning, and display the improved damping characteristics which result from scaling all eigenvalues of the system to the same order of magnitude.

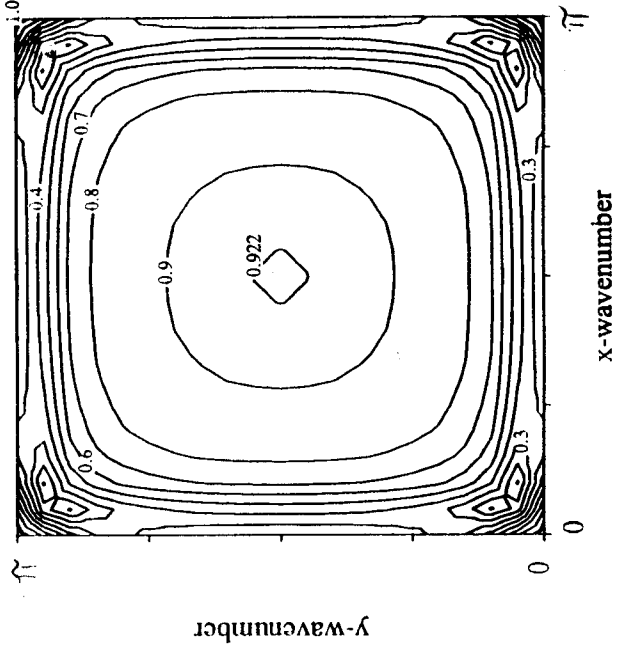


Fig. 2.1. Contours of the maximum amplification factor for the central-difference ADI scheme applied to the 2-D Euler equations. $M=0.5$, $CFL=5$, $v/u=1$.

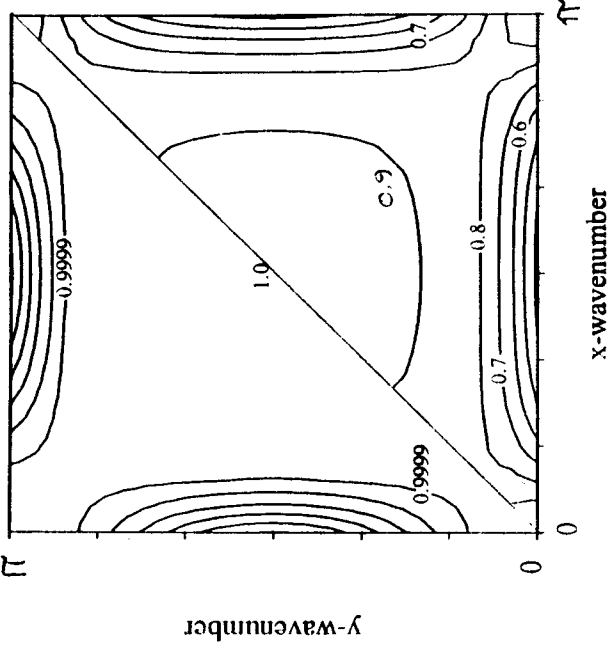


Fig. 2.2. Contours of the maximum amplification factor for the central-difference ADI scheme applied to the 2-D Euler equations. $M=0.01$, $CFL=5$. Upper triangle, standard equations (*i.e.* no preconditioning); lower triangle, with inviscid preconditioning.

ASSESSMENT OF PRESSURE-BASED ALGORITHMS

3.1 Introduction

Two widely used families of algorithms, the so-called ‘pressure-based’ and ‘density-based’ methods, have been developed for computational fluid dynamics problems over the years. Pressure-based methods include the Spalding-Patankar SIMPLE method [1], the MAC method [14] and the many derivatives of these. In this section, we focus on one of these derivatives, the PISO method [15, 16, 17]. In general, pressure-based procedures were formulated for incompressible flows at low Reynolds numbers, and were then extended to high Reynolds number flows and to compressible applications. They use an uncoupled, sequential solution of the equations of motion, and rely on diagonal dominance to obtain convergence. Pressure-based methods have employed finite volume formulations since their outset.

In order to examine the properties of the pressure-based schemes, the PISO algorithm will be written in a vector form which is similar to that commonly used for the density-based schemes. Some of the similarities and differences between the PISO pressure-based scheme and the preconditioned density-based schemes are highlighted, and vector stability analysis is applied to the PISO algorithm to ascertain its stability limits and convergence properties.

3.2 Vector Formulation of PISO Algorithm

The PISO algorithm is a time-dependent pressure-based scheme that is composed of a predictor step followed by two corrector steps [15]. In the predictor step, only the momentum equations are updated using the existing pressure and temperature fields. The first corrector step then begins by updating the pressure Poisson equation that enforces continuity. The Poisson equation is obtained by combining continuity and the divergence of the momentum equations. The first correction to the momentum equations is then made, followed by the first correction to the energy equation. The second corrector then repeats this Poisson-momentum-energy sequence again to complete a single time step.

The primary dependent variables that appear in the PISO scheme are the same as those contained in the vector, Q_n , except that the temperature is replaced by the total energy per unit mass, e/ρ . Because this change makes little difference, we base our formulation on the variable Q_v in which the temperature is the fourth variable.

The notation we employ for the predictor-corrector sequence differs from that used by Issa [15, 16]. To provide uniformity, we define predicted values for all four variables, even though only the velocity components are updated in the predictor step. (The predicted values of the pressure and temperature are set equal to their values at the old time step.) Predicted values are denoted by a single superscript *, while first corrector values

contain a double superscript **. The second corrector (which is the final value at the new time step) is denoted by a superscript $n + 1$. The value at the previous time step is denoted by superscript n .

In the PISO algorithm, the inviscid flux vectors are split into two parts,

$$E = E_L + E_N \quad \text{and} \quad F = F_L + F_N \quad (3.1)$$

where the subscripts L and N refer to a 'linear' and a 'nonlinear' part. (The linear term is linear in the vector Q , rather than the vector, Q_v .) The individual vectors are,

$$\begin{aligned} E_L &= (\rho u, p, 0, 0)^T, & E_N &= (0, \rho u^2, \rho uv, (e + p)u)^T \\ F_L &= (\rho v, 0, p, 0)^T, & F_N &= (0, \rho uv, \rho v^2, (e + p)v)^T \end{aligned} \quad (3.2)$$

Note the contrast between this flux splitting, and the standard Steger-Warming splitting which is based on the characteristics of the complete system of equations. The present splitting is chosen for convenience in solving the equations, while the previous one was chosen to mimic the physics of the unsteady flowfield. The philosophy of the splitting is to divide the flux vectors E and F in such a manner that their Jacobians can be written in a (lower or upper) triangular form that enables sequential solution. As will be seen, some approximation is required before triangular matrices are obtained, but the resulting sequential solution represents a substantially reduced operation count at each time step. The advantages of the sequential solution become more significant as the number of equations increases (as for example with reacting flows). The ultimate benefit of the uncoupling procedure; however, depends upon the impact that the approximations have on the convergence rate. Convergence rates are discussed later.

Even though the equations in the PISO algorithm are solved in sequential fashion, it is useful to express them in vector form. The resultant vector notation provides a better overall view of the complete PISO iterative system, and makes comparison with density-based methods easier. The predictor-corrector-corrector sequence for the compressible Navier-Stokes equations is:

Predictor

$$\left\{ I_{ce} + I_m \left[\Gamma + \Delta t \left(\frac{\partial}{\partial x} A_N^n + \frac{\partial}{\partial y} B_N^n - L_n \right) \right] \right\} \Delta Q^* = -\Delta t I_m \left[\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} - L_v(Q_v) \right]^n \quad (3.3)$$

First Corrector

$$\left\{ \Gamma + \Delta t \left[\left(\frac{\partial}{\partial x} A_L^* + \frac{\partial}{\partial y} B_L^* \right) + I_e \left(\frac{\partial}{\partial x} A_N^* + \frac{\partial}{\partial y} B_N^* - L_v \right) \right] \right\} \Delta Q^{**} = -\Delta t \left[\Gamma^* \Delta Q^* + \left(\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} \right)^* \right] \quad (3.4)$$

Second Corrector

$$\left\{ \Gamma + \Delta t \left[\left(\frac{\partial}{\partial x} A_L^{**} + \frac{\partial}{\partial y} B_L^{**} \right) + I_e \left(\frac{\partial}{\partial x} A_N^{**} + \frac{\partial}{\partial y} B_N^{**} - L_v \right) \right] \right\} \Delta Q^{***} = -\Delta t \left[\Gamma^{**} \Delta Q^{**} + \Gamma^* \Delta Q^* + \left(\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} \right)^{**} \right] \quad (3.5)$$

Here, the variables ΔQ^* , ΔQ^{**} , ΔQ^{***} are defined as $\Delta Q^* = Q_v^* - Q_v^n$, $\Delta Q^{**} = Q_v^{**} - Q_v^*$, and $\Delta Q^{***} = Q_v^{n+1} - Q_v^{**}$. The selection matrices, $I_m = \text{diag}(0, 1, 1, 0)$ and $I_{ce} = \text{diag}(1, 0, 0, 1)$, select the momentum equations and the continuity and energy equations respectively for the predictor step. while $I_e = \text{diag}(0, 0, 0, 1)$ selects the energy equation in the corrector step. These equations as written remain coupled. The method used to uncouple them is given below.

The Jacobians $A_L = \partial E_L / \partial Q_v$ and $B_L = \partial F_L / \partial Q_v$ are obtained by standard means:

$$A_L = \begin{bmatrix} u/RT & \rho & 0 & -\rho u/T \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ u & p & 0 & 0 \end{bmatrix}$$

with a similar expression for B_L .

In keeping with the linearizations used in the PISO scheme, the matrices for the nonlinear flux vectors are (non-unique) multiplicative relations such that $E_N = A_N Q_v$ and $F_N = B_N Q_v$. The matrix A_N is given by

$$A_N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \rho u & 0 & 0 \\ 0 & 0 & \rho u & 0 \\ \frac{eu}{p} & \rho u^2 + e & \rho uv & \frac{\rho u R}{\gamma - 1} \end{bmatrix}$$

and B_N is analogous.

In the PISO algorithm, the terms in the linear flux vectors are centrally differenced, while the ones in the nonlinear vectors are generally upwind differenced. When upwind differencing is chosen, the direction is selected according to the eigenvalues of the nonlinear flux vector which are the local velocity components. The acoustic speeds do not appear. This is an important difference between coupled and uncoupled solution methods.

The PISO scheme for incompressible flows is identical to the compressible formulation given above, except that the energy equation is not solved. Therefore, the last element/row of the flux vectors/Jacobians disappears, and the split flux vectors become, $E_L = (u, p, Q)^T$ and $E_N = (0, u^2, uv)^T$. Similarly, the selection matrices simplify to $I_m = \text{diag}(0, 1, 1)$ and $I_c = \text{diag}(1, 0, 0)$, and select the momentum equations and the continuity equation for the predictor step while the matrix I_e vanishes. The nonlinear flux vector is again upwind differenced in the direction of the local velocity, while the linear flux vector is central differenced. The final difference is that the time derivative matrix, Γ_e , becomes $\Gamma_e = (1/\beta, \rho, \rho)$ where β is a scaling parameter for the time step.

3.2.1 Uncoupling the PISO Algorithm

The variables in each of the steps given above (Eqs. (3.3-3.5)) are coupled, indicating that a sequential solution cannot be obtained. To uncouple them, some approximations must be made. The compressible PISO method relies primarily upon uncoupling the energy equation from the continuity and momentum equations so that a solution process analogous to that used for the incompressible equations can be employed.

Uncoupling the energy equation is effectively accomplished by approximating the equation of state by using lagged values for the temperature correction in all three steps of the continuity and momentum equations. The perfect gas relations for the three steps are, respectively,

$$\begin{aligned} p^* &= \rho^* RT^n \\ p^{**} &= \rho^{**} RT^* \\ p^{n+1} &= \rho^{n+1} RT^{**} \end{aligned} \quad (3.6)$$

This corresponds to making the density a function of only the pressure for the update of the continuity and momentum equations.

With this modification to the equation of state, the coefficient matrix of the time derivative on the left hand side of Eq. (2.1), Γ_e , is replaced by the approximate matrix for the PISO scheme, Γ_p ,

$$\Gamma_p = \begin{bmatrix} 1/RT & 0 & 0 & 0 \\ u/RT & \rho & 0 & 0 \\ v/RT & 0 & \rho & 0 \\ \frac{u^2+v^2}{2RT} + \frac{1}{\gamma-1} & \rho u & \rho v & \frac{\rho R}{\gamma-1} \end{bmatrix} \quad (3.7)$$

where the temperature values at the previous time level are treated as constants in the computation of the Jacobian. The time derivative in the continuity equation in Eq. (2.1) originally contained both a pressure derivative and a temperature derivative. This change removes the temperature derivative, leaving only the pressure term. This is clearly a further step towards making the equations more 'pressure-based'. The temperature derivatives in the momentum equations are also dropped in the approximation for Γ_p . Both these changes in Γ_p and the similar ones in A_L described below are applied only to the left hand sides of Eqs. (3.3) to (3.5). The right hand sides remain unchanged, thereby ensuring formal temporal accuracy.

Again using the modified equation of state, and treating lagged values of temperature as constants in the determination of the Jacobian of the linear flux vector, A_L , we obtain the PISO approximate matrix, A_{Lp} :

$$A_{Lp} = \begin{bmatrix} u/RT & \rho & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ u & p & 0 & 0 \end{bmatrix} \quad (3.8)$$

This approximation again corresponds to dropping the temperature dependence in the continuity equation and completes the uncoupling of the energy equation (T) from the other equations. The continuity and momentum equations can now be updated separately from the energy equation.

After the energy equation has been uncoupled from the other equations, an uncoupled expression for the pressure is obtained by combining the discretized continuity and momentum equations. This step is identical to forming the pressure Poisson relation, except that it is done algebraically with the discretized equations. This ensures that continuity is satisfied in its discrete form. Once the new value for pressure is known, values for the velocity components can be obtained from their respective equations of motion. Having obtained new values of p , u , v , and ρ , the energy equation is then solved for the temperature using the latest values of these variables.

The approximations listed here enable the equations to be uncoupled from each other. The benefits of the sequential solution; however, cannot be realized if the approximations have an overly adverse effect on the convergence rate. As a means of assessing how the PISO uncoupling procedure affects convergence, we look at the vector stability characteristics of both the pressure- and density-based schemes.

3.2.2 Diagonally Dominant PISO Scheme

To enhance convergence in the PISO scheme, the central (diagonal) element of the explicit operator is evaluated at the new rather than the old time step. Because the central element does not appear when centered differencing is used, this is effective only for upwind differencing. This evaluation of explicit terms at the new (implicit) level, which is equivalent to adding an additional time derivative to the equation, provides considerable stability to the algorithm as noted below.

To demonstrate the procedure, we consider the simple scalar equation,

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (3.9)$$

Euler explicit differencing of this equation gives,

$$u^* - u^n + \lambda (u_i^n - u_{i-1}^n) = 0 \quad (3.10)$$

where λ is the Courant number, $\lambda = a\Delta t/\Delta x$. It is well known that this differencing is stable for Courant numbers, λ , less than unity.

Evaluating the central term of Eq. (3.10) at the implicit level gives

$$u^* - u^n + \lambda (u_i^* - u_{i-1}^n) = 0. \quad (3.11)$$

or, upon rearranging,

$$u^* - u^n + \lambda (u_i^* - u_i^n) + \lambda (u_i^n - u_{i-1}^n) = 0. \quad (3.12)$$

Thus, Eq. (3.12) is equivalent to the finite difference expression that is obtained when Euler explicit differencing is applied to the modified equation,

$$(1 + \lambda) \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (3.13)$$

Stability analysis shows that Eq. (3.13) is unconditionally stable for Euler explicit differencing, although the time step for infinite Courant number is identically equal to the time step obtained with a Courant number of unity in Eq. (3.10).

We can use this same modification to the PISO scheme in Eqs. (3.3-3.5) to denote the evaluation of the central terms at the new time level. Thus, we define the matrix, Γ , in the corrector steps. Eqs. (3.4) and (3.5), as

$$\Gamma = \Gamma_e + A_N \frac{\Delta t}{\Delta x} \quad (3.14)$$

This gives a diagonally dominant form of the equations of motion which is more stable than is the original form.

3.3 PISO Stability Results

3.3.1 Conditional Stability of the PISO Scheme

Although the PISO algorithm allows the equations to be solved separately, the solution variables remain coupled to each other. For example, an error in the pressure at a particular time step will propagate to the velocity and temperature solutions in succeeding steps. Thus the analysis that is used to assess the stability and convergence of the PISO method cannot be based on the individual equations, but must include a vector stability analysis of the complete equation system. With the equations written in vector form, Eqs. (3.3-3.5), the stability analysis follows by standard methods. The PISO results are for first-order upwind differencing of the convective terms.

Figure 3.1 shows the maximum eigenvalue at a Courant number of one for the compressible PISO algorithm at Mach 0.3. As can be seen, the PISO scheme is quite stable at this Courant number except in the low wave number region. The amplification factors at low wave numbers indicate a weak instability that is essentially independent of Mach number and persists even in the incompressible limit. Numerical experiments shown later demonstrate that this instability can lead to divergence in computational solutions. Surprisingly, this instability disappears when central differencing is used for the convective terms.

At higher values of the Courant number, the approximations used to uncouple the energy equation begin to take their toll, and the PISO scheme begins to exhibit high wave

number instabilities. Representative results for the uncoupled form of the compressible PISO formulation are shown on the lower portion of Fig. 3.2 for a Courant number of 5. The high wave-number instability shown here is present only in the compressible equations whereas (as noted before) the low wave-number instability occurs in both the compressible and the incompressible equations. Comparisons with stability calculations for the incompressible PISO formulation indicate that the high wave-number instability arises from the treatment of the energy equation. Stability results for the compressible system with Γ_p replaced by Γ_e (so that the equations are once again coupled), reveal that stability is regained when the coupling between the continuity and energy equations is retained in the time derivative. Similar calculations with the Jacobian matrix, A_{Lp} replaced by the exact Jacobian, A_L , show very little effect on the amplification factor. Thus, it appears that omitting the energy coupling term in the time derivative has an adverse effect on the maximum allowable time step, but the approximation in the spatial derivatives is relatively insignificant. Nevertheless, we note that, for low speeds, the stability characteristics at the smaller time steps (CFL=1) provide excellent amplification factors which should translate into good convergence if the time step is chosen appropriately. The high wave-number instability; however, causes the maximum allowable Courant number to decrease as the Mach number approaches transonic speeds. This clearly suggests the importance of treating the energy coupling terms carefully at high speeds where compressibility becomes more dominant.

A peculiarity associated with the uncoupled PISO method is that although the overall algorithm is conditionally stable for large time steps, the individual steps in the predictor-corrector sequence are all unconditionally stable. If we define the amplification factors G^* , G^{**} , and G^{***} for the predictor and the first and second corrector steps so that $Q^* = G^*Q^n$, $Q^{**} = G^{**}Q^*$, and $Q^{n+1} = G^{***}Q^{**}$, analysis shows that all eigenvalues of these three amplification matrices are less than unity. Upon combining the three steps and defining the overall amplification matrix as the product of the three steps, $Q^{n+1} = GQ^n$, or $G = G^{***}G^{**}G^*$, an analogous stability analysis shows that some eigenvalues of G are greater than unity at selected wave numbers (see Fig. 3.2). This unexpected result, that the product of three stable steps is unstable, is a clear demonstration of the differences between scalar equations and systems of equations. Multi-step amplification factors for scalar equations are stable if the amplification factors of all the sub-steps are stable, but this generalization does not extend to amplification matrices. The only time this simplification can be applied to systems of equations is when the amplification matrices of the individual steps can be diagonalized by the same similarity transformation—when they have the same eigenvectors. In the present case, the amplification matrices for the two corrector steps have the same eigenvectors, but the eigenvectors for the predictor step are different. It is for this reason that the product of three steps that are in themselves stable can be unstable.

Comparisons of stability prediction and actual numerical convergence for the one-dimensional, incompressible PISO scheme are shown in Figure 3.3. Periodic boundary conditions were specified for the numerical runs and pure-mode errors were introduced as initial conditions. The growth or decay of these errors was monitored and the numerical amplification factors were determined. These numerical amplification factors are plotted as the symbols in Fig. 3.3a. The solid line represents the spectrum of amplification factors as determined by the stability analysis. Comparison between the analysis and the numerical

results is excellent, verifying the results of the stability analysis. Figure 3.3b shows that actual convergence for three cases, each with a different initial error (wave-numbers of 1, 3, and 5). The numerical convergence is plotted as the solid line, and the anticipated convergence (determined from the stability analysis) is shown as the dashed lines. Both the actual and anticipated convergence histories fall very close to each other.

The effect of inflow/outflow boundaries on the stability of the incompressible PISO scheme are shown in Fig. 3.4. The initial condition is a one point, 10% perturbation of the exact solution. The convergence histories for CFL's from 1 to 12 are shown. The convergence improves as the CFL is increased from 1 to 8; however, at the higher CFL's (2, 4 and 8) the convergence makes an abrupt change in slope as the least stable error mode dominates the field of errors. At CFL=10 the algorithm should be unstable, but converges, albeit at a slower rate than the CFL=1 case. This is due to the errors propagating out of the domain before they can grow enough to contaminate the solution. Finally, at CFL=12 the errors grow fast enough such that the solution diverges.

3.3.2 PISO At Different Mach Numbers

The effect of the Mach number on the stability characteristics of the 2-D compressible PISO scheme is also examined. Figure 3.1 displayed the amplification factors for the fully implicit PISO scheme at a Mach number of 0.3. The amplification factors for a Mach number of 0.03 are shown in Fig. 3.5, and indicate that the PISO scheme actually becomes more stable as the Mach number is decreased. This is consistent with the commonly observed behavior of pressure-based schemes, namely, that their convergence rate is not adversely affected for low speed compressible flows.

These favorable characteristics at low Mach numbers appear to diminish when an approximate solution method such as the ADI scheme is used to efficiently solve the two-dimensional system. Figure 3.6 shows amplification factors for Mach 0.3 and 0.03 using the PISO scheme with ADI factorization. Now the algorithm is stiff at the Mach 0.03 condition as opposed to Figure 3.5, which showed very good damping. The reason for the stiffness at low Mach numbers is because of non-optimal time stepping for acoustic propagation, which leads to large approximate factorization errors. This stiffness can easily be removed by simply scaling the time derivative in the continuity equation by the Mach number squared. Figure 3.7 shows the effect of scaling the time derivative for Mach 0.3 and 0.03. The PISO-ADI scheme now exhibits damping which is essentially independent of the Mach number.

The line Gauss-Seidel relaxation method can also be used to efficiently solve the 2-D PISO system, and the stability characteristics of this approach are shown in Figs. 3.8 and 3.9, again for Mach numbers of 0.3 and 0.03. Figure 3.8 shows the amplification factors after only one relaxation sweep of the line Gauss-Seidel procedure. Here, the stability maps look considerably different from those of the direct inversion shown in Figs. 3.1 and 3.5. In fact, the Mach 0.03 case is seen to be somewhat unstable to low wave-number modes (Fig. 3.8). However, after 10 relaxation sweeps of the line Gauss-Seidel procedure both the Mach 0.3 and 0.03 stability maps (Fig. 3.9) look identical to the direct inversion (Figs. 3.1 and 3.5), suggesting good convergence.

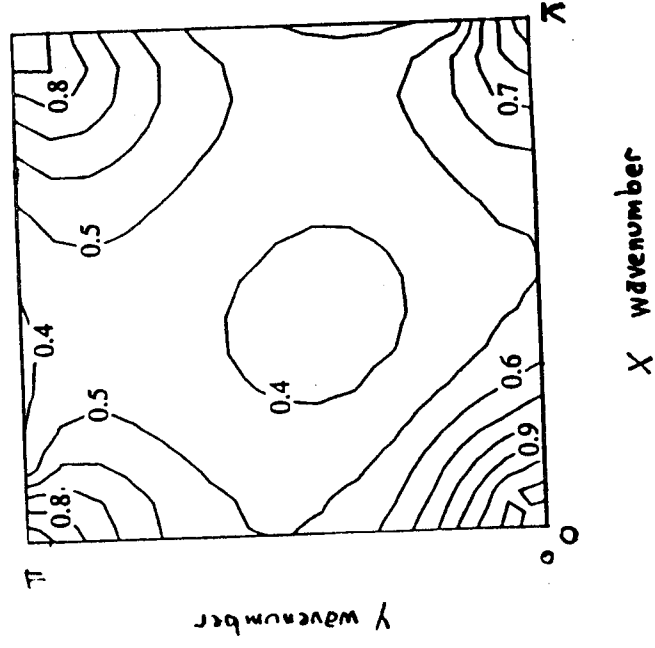


Fig. 3.1. Contours of the maximum amplification factor for compressible PISO algorithm at $M=0.3$, $CFL=1$.

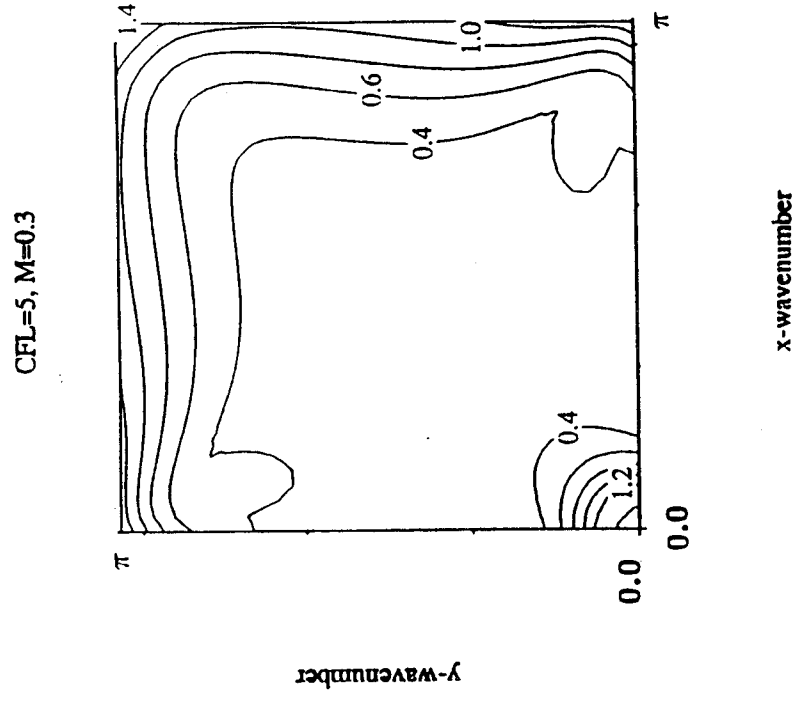


Fig. 3.2. Contours of the maximum amplification factor for compressible PISO algorithm at $M=0.3$, $CFL=5$.

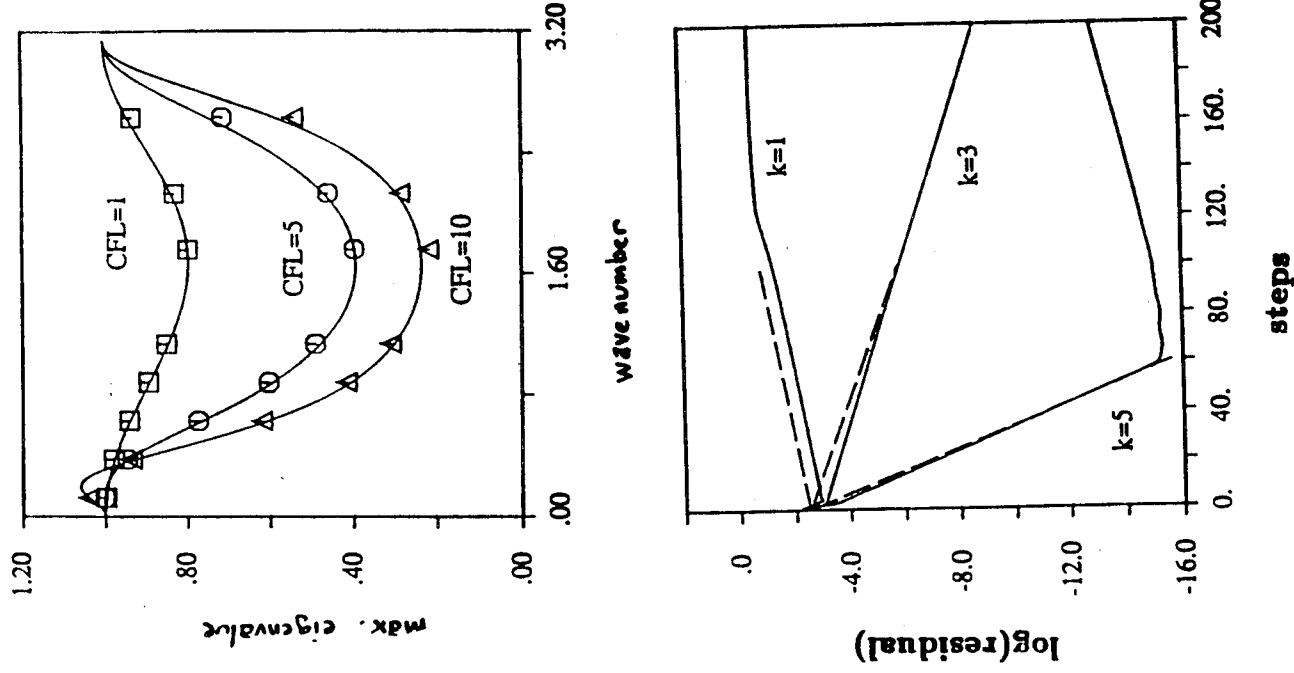


Fig. 3.3. Comparison between stability predictions and actual numerical runs. (a) Amplification factors from stability analysis and from actual numerical convergence for the 1-D, incompressible PISO scheme applied to the Euler equations. Solid lines are the stability prediction, and the symbols are from actual numerical convergence. (b) Convergence plots for three pure-mode errors (wave-numbers of 1, 3 and 5). Dashed lines are predicted convergence from stability, and solid lines are actual numerical convergence.

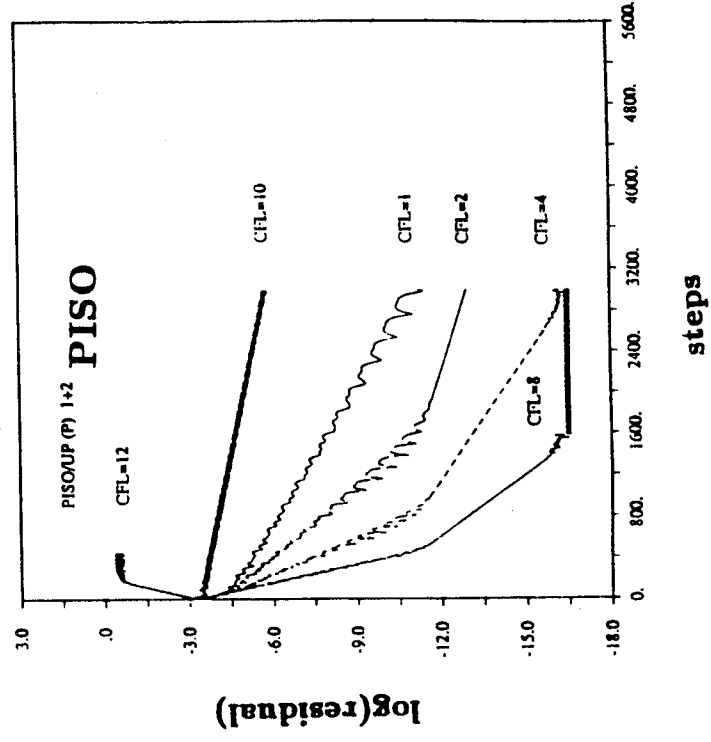


Fig. 3.4. Effect of inflow/outflow boundaries on the stability and convergence of the 1-D, incompressible PISO scheme.

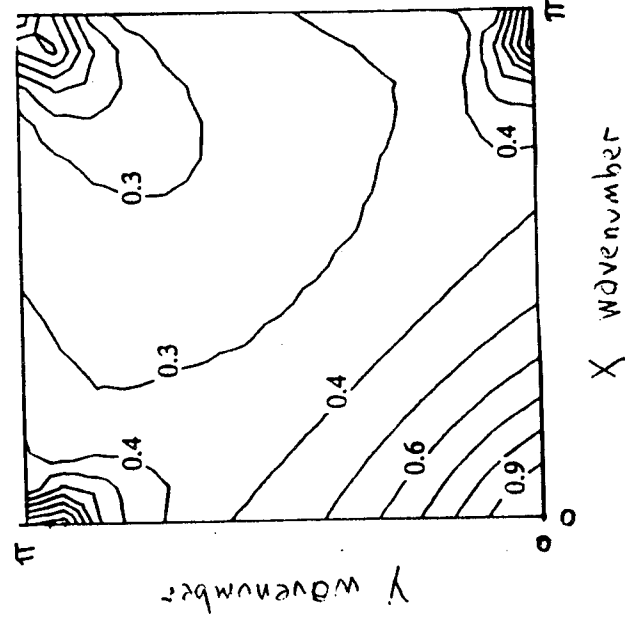


Fig. 3.5. Contours of the maximum amplification factor for compressible PISO algorithm at $M=0.03$, $CFL=1$.

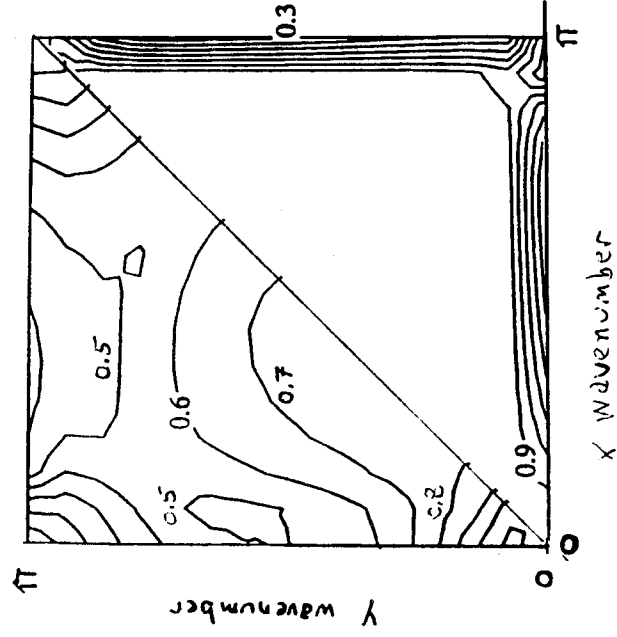


Fig. 3.6. Contours of the maximum amplification factor for compressible PISO algorithm using ADI factorization, CFL=1. Upper triangle, $M=0.3$; lower triangle, $M=0.03$.

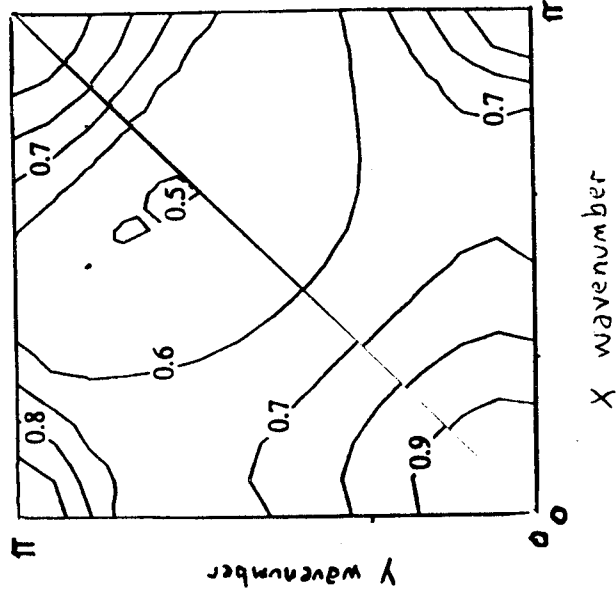


Fig. 3.7. Contours of the maximum amplification factor for compressible PISO algorithm using ADI factorization but with the time derivative of the continuity equation scaled by M^2 , CFL=1. Upper triangle, $M=0.3$; lower triangle, $M=0.03$.

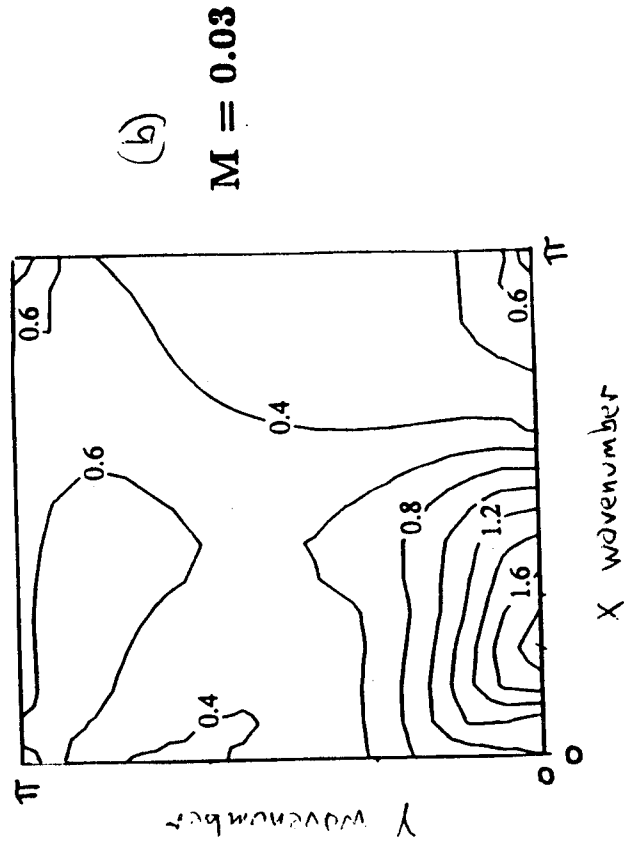
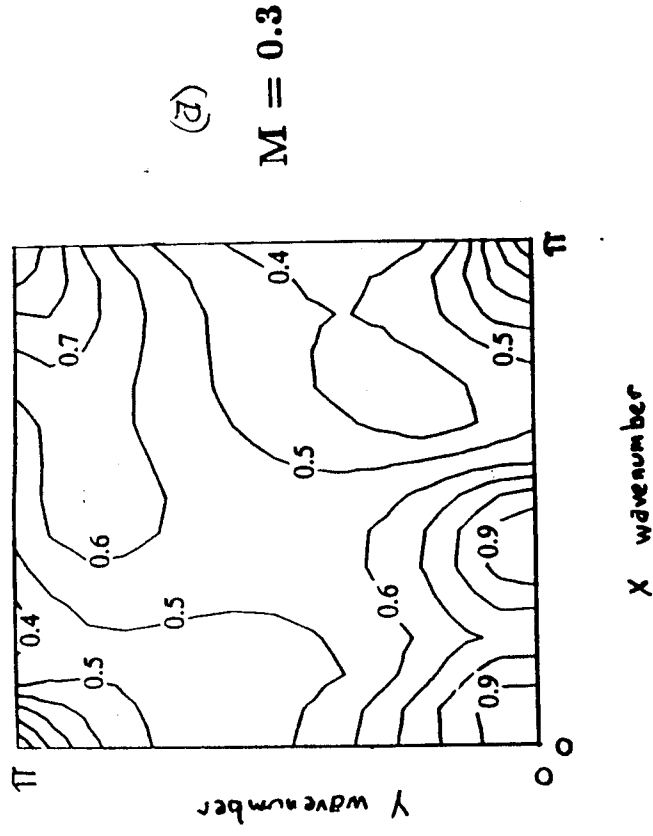


Fig. 3.8. Contours of the maximum amplification factor for compressible PISO algorithm using Line Gauss-Seidel (LGS) relaxation and one relaxation sweep, CFL=1. (a) $M=0.3$, (b) $M=0.03$.

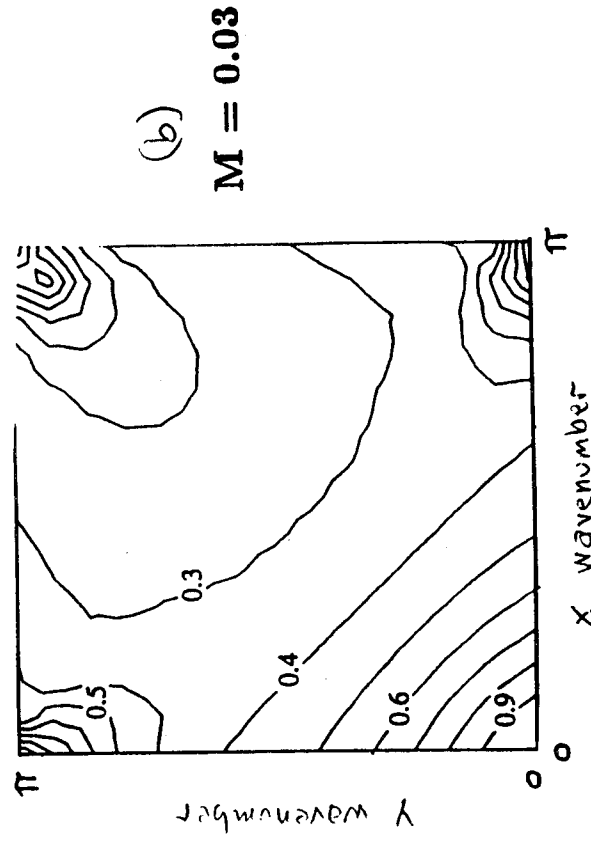
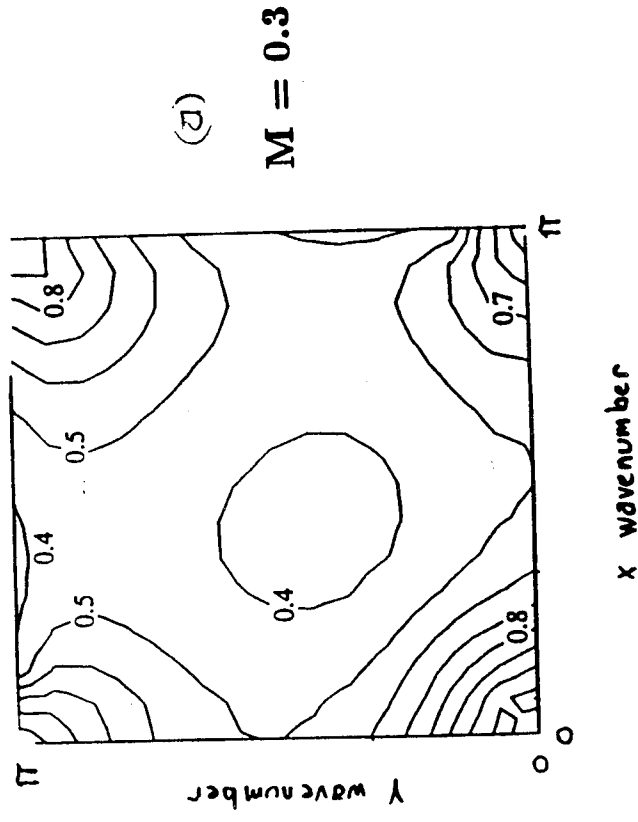


Fig. 3.9. Contours of the maximum amplification factor for compressible PISO algorithm using Line Gauss-Seidel (LGS) relaxation and 10 relaxation sweeps, CFL=1. (a) $M=0.3$, (b) $M=0.03$.

VISCOUS PRECONDITIONING

4.1 The 1-D Navier Stokes Equations

The unsteady Navier-Stokes equations are a mixed hyperbolic-parabolic set. In the high Reynolds number limit, the equations are hyperbolic and the eigenvalues of the system represent the speeds of propagation of the disturbances. At low Reynolds numbers, the viscous terms start becoming important and cause damping of the disturbances in the flowfield. The interaction of the hyperbolic wave propagation and parabolic viscous damping processes is complicated and is crucial to the proper selection of the preconditioning matrix. The inviscid procedure of conditioning of the wave speeds of the system is no longer physically relevant. In particular, it is necessary to simultaneously optimize the relevant propagation velocities and the viscous damping rates.

To gain a better understanding of the physical characteristics of the Navier-Stokes equations over a broad range of Reynolds numbers, it is useful to look for an approximate analytical solution. To do so, we start from the one-dimensional equations,

$$\Gamma_e \frac{\partial Q_v}{\partial t} + \frac{\partial E}{\partial x} = \frac{\partial}{\partial x} R_{xx} \frac{\partial}{\partial x} Q_v \quad (4.1)$$

where, Γ , Q_v , E and R_{xx} are the one-dimensional versions of the flux vectors given in Eqn. (2.1). We seek a closed-form solution of the form $Q_v(x, t) = \hat{Q}_v e^{i\omega t} e^{-ikx}$ for the linearized constant-coefficient approximation of Eqn. (4.1). Substituting into the governing equation, we obtain a general dispersion relation linking the complex frequency ω and the wave number k ,

$$\det [\Gamma_e(i\omega) - A(ik) + R_{xx}(k^2)] = 0 \quad (4.2)$$

where A is the inviscid Jacobian $\frac{\partial E}{\partial Q_v}$. For the case of $Pr = 1$ and neglecting viscous dissipation, we can obtain the following roots for the complex frequency.

$$\omega_{1,2,3}/k = u(1 + i/Re) \quad \omega_{2,3}/k = u \left[1 - \frac{i\gamma}{2Re} \left(-1 \pm \sqrt{1 - \frac{4Re^2}{\gamma^2 M^2}} \right) \right] \quad (4.3)$$

where the Reynolds number is defined using $1/k$ as the length scale, i.e., $Re = u/k\nu$.

In the limit of high Reynolds number, the first expression becomes $\omega_1/k = u$, while the second one reduces to $\omega_{2,3}/k = u \pm c$, which are the familiar inviscid results. In particular, we note that the first expression provides the particle eigenvalue, while the second, more complex expression furnishes the acoustic eigenvalues.

When the flow is viscous-dominated and the Reynolds number is low, two possible situations may arise depending on the magnitude of a second parameter, the “acoustic” Reynolds number, $Re_c = c/\nu k = Re/M$. When the acoustic Reynolds number is high (which happens for low Mach number, low Reynolds number situations), the first expression yields an imaginary root, $\omega_1/k = iu/Re$, while the second expression remains real, $\omega_{2,3}/k = u \pm c$. The first root indicates viscous damping of the particle wave, while the second and third roots indicate acoustic propagation at $u + c$ and $u - c$. This result suggests that acoustic wave propagation processes remain dominant under low Mach number, low Reynolds number conditions, even though the viscous processes dominate the other modes of the solution.

The second situation that arises in the limit of low Reynolds number corresponds to when the acoustic Reynolds number is also small in comparison to unity, i.e., when Re/M is small. In this case, the first root remains imaginary, $\omega_1/k = iu/Re$; however, the term under the square-root in the second expression now tends to unity—the root with the negative sign then gives a complex root ($\omega_2/k = iu\gamma/Re$), while the root with the positive sign remains real, $\omega_3/k = u$. The first two roots then correspond to viscous damping modes while the third root indicates a wave that propagates at the particle velocity. The two imaginary roots apparently arise from the viscous terms in the momentum and energy conservation equations, while the real root corresponds to the continuity equation (which does not contain any viscous terms). This apparent ‘de-coupling’ of the governing equations into one wave equation and a set of heat diffusion equations occurs because the pressure gradient term in the momentum equation becomes negligible compared to the viscous term at low acoustic Reynolds numbers.

From this simple analysis, it is apparent that the behaviour of the Navier-Stokes equations is rich in complexity. The controlling physical processes vary significantly in character depending on the Mach number and the particle and acoustic Reynolds numbers. In particular, one of the inviscid modes (corresponding to wave propagation) remains relevant even when the problem is viscous-dominated. The results for the different limiting cases are summarized in Table 4.1, which also gives the associated condition numbers of the system. The condition number is an indication of the stiffness of the system and is defined as:

$$\sigma = \frac{\text{Max}(|\omega_1/k|, |\omega_2/k|, |\omega_3/k|)}{\text{Min}(|\omega_1/k|, |\omega_2/k|, |\omega_3/k|)}$$

When σ is near unity, the system is well-conditioned and when it is much larger than unity, the system becomes stiff and poor convergence is observed. From Table 4.1, it is clear that the standard Navier-Stokes equations become stiff at low Mach numbers and/or at low Reynolds numbers.

Preconditioning the time-derivative is a potential method of controlling the transient characteristics and, thereby, improving the conditioning of the system. From the above results, it is clear that, for the Navier-Stokes equations, the preconditioning methodology should possess three properties—(1) conditioning of the acoustic and particle wave-speeds at high Reynolds numbers (or inviscid preconditioning), (2) optimizing the acoustic speeds with the viscous damping rate at low particle Reynolds numbers/high acoustic Reynolds numbers, and (3) optimizing the velocity scale of the continuity equation with the viscous damping

rates at low particle Reynolds/low acoustic Reynolds numbers. The present preconditioning strategy, which involves scaling the time-derivative of pressure in the continuity equation shows promise in this respect because it provides a means of controlling either the acoustic speeds or the time-scale of the continuity equation whichever is relevant. We next consider the characteristics of the preconditioned system under viscous-dominated conditions.

4.2 The Preconditioned Navier-Stokes Equations

The preconditioned system is different from the original system in that the matrix Γ (Eqn. 2.2) is replaced by the preconditioning matrix Γ_p (Eqn. 2.3). The dispersion relation for the preconditioned system is identical to the one given in Eqn. (4.2) except that the matrix Γ_p is used. The roots of the complex frequency for the this modified system are given by:

$$\omega_{2,3}/k = \frac{u}{2} \left\{ \left[(1+\epsilon) + i/Re \right] \pm \sqrt{(1-\epsilon)^2 + 4\epsilon/M^2 - 1/Re^2 + 2i/Re[1 - \epsilon(2\gamma - 1)]} \right\} \quad (4.4)$$

The above roots are more complicated than the roots of the original equations. Nevertheless, the desired physical insight may be gained by considering the limiting behaviour of the system under low Reynolds number situations. In particular, the roots enable us to select ϵ so as to maintain good conditioning of the system under different regimes.

As discussed earlier, the choice of $\epsilon = 1$ corresponds to the non-preconditioned limit of the Euler equations. For the Navier-Stokes equations, this choice yields characteristics that are similar to (but not identical with) the standard equations. This is clear from Table 4.2, where the limiting behaviour of Eqn. (4.4) with $\epsilon = 1$ is summarized.

Preconditioning the Euler equations involves selecting $\epsilon = M^2$ for subsonic flows as given in Eqn. (2.5). Table 4.3 summarizes the limiting behaviour of Eqn. (4.4) for this choice of ϵ . For high Reynolds numbers, as anticipated, this inviscid preconditioning significantly improves the conditioning of the system. However, at lower Reynolds number, the condition number becomes progressively larger, increasing as $1/Re^2$. In fact, the condition number at low Reynolds numbers is actually worse than that of the original system. This is because the inviscid preconditioner is selected based on the particle and acoustic wave speeds of the system, which are not physically relevant in the viscous limit. Hence, an alternate preconditioning strategy is necessary when the Reynolds number is low.

4.3 Formulation of Viscous Preconditioning

It is necessary to formulate a preconditioning methodology that is optimal for all Mach numbers *and* all Reynolds numbers. The approach that we take here is to retain the form of the preconditioning matrix Γ_p as defined in Eqn. (2.3) and select the parameter ϵ in this matrix so that the relevant physical processes are optimized. The definition of ϵ in the inviscid limit has been previously discussed. In this section, we focus on the definition of ϵ in the viscous limit and then we generalize this definition for all Mach numbers and Reynolds numbers.

We have discussed the characteristics of the Navier-Stokes equations in a previous section. At low particle and high acoustic Reynolds numbers, the dominant processes are viscous damping and acoustic wave propagation. The time scales associated with these disparate processes may be equalized by choosing ϵ to be M_r^2/Re^2 . Note that this choice of ϵ scales the pressure time-derivative and, hence, the acoustic speed by the viscous damping rate. Table 4.4 shows the limiting behaviour of the roots of Eqn. (4.4) for this choice of ϵ . The condition number is now identically unity, a result that is obtained as long as Re/M^2 is much higher than unity. Note that this latter condition is based on a new quantity Re/M^2 and not on the acoustic Reynolds number, Re/M . This altered criterion appears to be a facet of this particular choice of preconditioning matrix.

As discussed earlier, a second situation arises when both the particle and acoustic Reynolds numbers are low. The dominant scales are now the viscous damping rate and the velocity scale of the continuity equation (i.e., the fluid velocity). Now, the appropriate choice of ϵ becomes $1/Re$, which effectively scales the time scale of the continuity equation to be the same as the momentum and energy equations. Table 4.4 again summarizes the limiting behaviour of Eqn. (4.4) for this choice of ϵ . The associated condition number is 1.18 as long as Re/M^2 is much less than unity. Note that this criterion is based again on the new quantity Re/M^2 and not on the acoustic Reynolds number and is consistent with the earlier observation.

The results in Table 4.4 suggest the trends that the definition of the preconditioning parameter, ϵ , should follow for optimal conditioning of the system; however, it does not take into account the transition regions between the limits considered. One definition of ϵ which approaches M^2/Re^2 when Re/M^2 is large and approaches $1/Re$ when Re/M^2 is small is given by:

$$\epsilon_{\text{vis}} = \frac{M^2 (\alpha - 1) \alpha}{1 + M^2 (\alpha - 1)} \quad (4.5)$$

where $\alpha = 1/Re$. Since the above definition ensures proper behaviour of ϵ in the viscous limit, we designate it as ϵ_{vis} . Unification of Eqn. (4.5) with the definition for the inviscid limit is considered later.

The definition of ϵ_{vis} as given in Eqn. (4.5) introduces two difficulties. First, Eqn. (4.5) depends on the wavenumber k (recall that $Re = u/k\nu$). Clearly, in the context of a CFD code, ϵ should be single-valued. Therefore, in practice, ϵ may be selected based on the optimal conditioning of only a single wave-mode. Almost all numerical schemes possess good damping characteristics for the higher modes either through inherent or artificial dissipation. Thus, for the discrete Navier-Stokes equations, it is the low wavenumbers that typically control convergence rates and the tuning of the preconditioning procedure to these modes maximizes the benefit. Therefore, ϵ is defined based on the lowest wavenumber, which corresponds to using the cell-Reynolds number ($\alpha = 1/Re_{\Delta x} = \nu/u\Delta x$) in Eqn. (4.5).

The second problem with Eqn. (4.5) is that it equates the time-scales of the wave-propagation modes to those of the viscous damping modes. For most numerical algorithms, however, the limiting or optimal time-scales of these processes, represented by the CFL number ($\lambda\Delta t/\Delta x$) and the von Neumann number ($VNN = \nu\Delta t/\Delta x^2$), are not in general equal. For instance, for the explicit four-stage Runge-Kutta scheme, the limiting CFL number is

$2\sqrt{2}$ while the limiting VNN is around 0.25. For the implicit ADI scheme, on the other hand, the optimal CFL number is between 3 and 5, while the optimal VNN is between 10 and 100. If Eqn. (4.5) were used to define ϵ , the CFL and VNN numbers of the system would have equal magnitudes, which would be non-optimal under some circumstances. In order to account for these effects properly, we choose ϵ_{vis} by equating the physical time-step size obtained from the CFL and von-Neumann numbers. This amounts to rescaling α in Eqn. (4.5) so that $\alpha = CFL/(VNN \cdot Re_{\Delta x})$.

At low Reynolds numbers, Eqn. (4.5) specifies ϵ_{vis} to be greater than M^2 . As the Reynolds number approaches unity, ϵ_{vis} approaches M^2 , the inviscid value of this parameter. As the Reynolds number is further increased, ϵ_{vis} becomes equal to zero and then takes on negative values. Clearly, this is inappropriate. Therefore, it is necessary to limit the minimum value of the preconditioning parameter to the proper inviscid value as follows:

$$\epsilon = \text{Max} (\epsilon_{\text{inv}}, \epsilon_{\text{vis}}) \quad (4.6)$$

where ϵ_{inv} is given by Eqn. (2.5) and ϵ_{vis} is given by Eqn. (4.5). The above definition maintains the system well-conditioned at all Mach numbers and Reynolds numbers by optimizing the dominant physical processes. Figure 4.1 shows the variation of ϵ in the Mach number/Reynolds number domain as given by Eqn. (4.6). It is evident that the value of ϵ follows the trends summarized in Table 4.4.

For multi-dimensional computations, similar expressions may be obtained by employing the controlling scales in the different co-ordinate directions and picking the most restrictive condition for the definition of ϵ_{vis} :

$$\epsilon_{\text{vis}} = \text{Max} \left[\frac{\alpha(\alpha - 1)}{(\alpha - 1 + c^2/u^2)}, \frac{\beta(\beta - 1)}{(\beta - 1 + c^2/v^2)} \right] \quad (4.7)$$

where $\alpha = CFL/(VNN \cdot Re_{\Delta x})$ and $\beta = CFL/(VNN \cdot Re_{\Delta y})$. The definition of ϵ_{inv} remains unchanged. In the presence of highly stretched grid cells, Eqn. (4.7) is no longer effective. High grid aspect ratio issues are dealt with later.

4.4 Von Neumann Stability Results

Von Neumann stability theory is a straightforward means of ensuring that the viscous preconditioning matrix indeed ensures good conditioning in the presence of low cell Reynolds numbers. Figure 4.2 shows the stability diagram as a function of wavenumber space for the two-dimensional preconditioned Navier-Stokes equations. Stability results are shown for $M = 0.001$ and $Re_{\Delta} = 0.001$ with inviscid preconditioning ($\epsilon = \epsilon_{\text{inv}}$) and viscous preconditioning ($\epsilon = \epsilon_{\text{vis}}$). In the first case, when inviscid preconditioning is used, the maximum amplification factor of the system is observed to be near unity over the entire wavenumber domain, indicating that the system is extremely stiff. This result is in accordance with the results summarized in Table 4.3.

When viscous preconditioning is used, for this low Reynolds number case, the situation is dramatically different. The maximum amplification factor is now significantly less than unity over most of the wavenumber domain. Particularly in the low wavenumber region,

which is the region that the preconditioning procedure is tuned for, very good amplification factors are observed. The slight stiffness in the high wavenumber regions is not of much significance since the addition of artificial dissipation tackles these higher modes very effectively. (The stability diagram is for the case without artificial dissipation). Thus, the viscous preconditioning procedure promises good stability and convergence behaviour when the cell Reynolds number is small.

4.5 Computational Results

In order to demonstrate the effectiveness of the preconditioning formulation for Navier-Stokes problems, we consider the simple example of developed flow in a 2-D channel. The aspect ratio of the channel geometry is unity (length equal to height) and the grid size employed is 41×41 . In order to assess the performance of the preconditioned system to different flow conditions, we vary the flow Mach number (by changing mean gas velocity) and the flow Reynolds number (by changing the gas viscosity). The convergence results from this study are plotted in Figs. 4.3 and 4.4 which show the number of iterations required for convergence to machine accuracy versus the Mach number (Fig. 4.3) and the cell Reynolds number (Fig. 4.4). Figure 4.3 shows the effectiveness of the inviscid preconditioner since the results are for $Re_{\Delta x} = 1000$ which is basically inviscid flow. The results show that at low Mach numbers, the original equations yield extremely poor convergence while, with preconditioning, attractive convergence rates (less than 1000 steps) are maintained down to Mach numbers of 10^{-5} .

Figure 4.4 shows the convergence rates as a function of Reynolds number at a fixed Mach number of 0.1. Using just inviscid preconditioning, good convergence rates are obtained only until the cell Reynolds number is about 1. For lower cell Reynolds numbers, convergence deteriorates drastically (very much like the original system does at low Mach numbers). On the other hand, with the viscous preconditioning procedure described in this paper, attractive convergence rates (consistently less than 1000 steps) are maintained over the whole range of Reynolds numbers. These results demonstrate that preconditioning the equations of motion based on the flow conditions can improve algorithm convergence by several orders of magnitude thus enabling a major savings in computer time and expense. We consider more realistic flowfield examples in the following section after taking into account the effect of high cell aspect ratios on algorithm convergence.

Reynolds no. limits	ω_1/k	ω_2/k	ω_3/k	condition #
$Re \gg 1$	u	$u + c$	$u - c$	$1 + 1/M$
$Re \ll 1, Re/M \gg 1$	$\frac{i u}{Re}$	$u + c$	$u - c$	$(1 + 1/M) Re$
$Re \ll 1, Re/M \ll 1$	$\frac{i u}{Re}$	u	$\frac{i u \gamma}{Re}$	γ/Re

Table 4.1. Original Navier-Stokes

Reynolds no. limits	ω_1/k	ω_2/k	ω_3/k	condition #
$Re \gg 1$	u	$u + c$	$u - c$	$1 + 1/M$
$Re \ll 1, Re/M \gg 1$	$\frac{i u}{Re}$	$u + c$	$u - c$	$(1 + 1/M) Re$
$Re \ll 1, Re/M \ll 1$	$\frac{i u}{Re}$	$\frac{i u}{Re}$	$\gamma u + \left(\frac{i u Re}{M^2}\right)$	$\sqrt{\gamma^2 Re^2 + Re^4/M^4}$

Table 4.2. Preconditioned Navier-Stokes, $\epsilon=1$

Reynolds no. limits	ω_1/k	ω_2/k	ω_3/k	condition #
$Re \gg 1$	u	$\frac{u}{2}(1 + \sqrt{5})$	$\frac{u}{2}(1 - \sqrt{5})$	2.61
$Re \ll 1, Re/M \gg 1$	$\frac{i u}{Re}$	$\frac{i u}{Re}$	$\frac{5}{4}(i u Re)$	$\frac{5}{4} \frac{1}{Re^2}$
$Re \ll 1, Re/M \ll 1$	$\frac{i u}{Re}$	$\frac{i u}{Re}$	$\frac{5}{4}(i u Re)$	$\frac{5}{4} \frac{1}{Re^2}$

Table 4.3. Preconditioned Navier-Stokes, $\epsilon = M^2$

Reynolds no. limits	ω_1/k	ω_2/k	ω_3/k	condition #
$Re \gg 1 (\epsilon = M^2)$	u	$\frac{u}{2}(1 + \sqrt{5})$	$\frac{u}{2}(1 - \sqrt{5})$	2.61
$Re \ll 1, Re/M \gg 1, (\epsilon = \frac{M^2}{Re^2})$	$\frac{i u}{Re}$	$\frac{u}{2Re}(i + \sqrt{3})$	$\frac{u}{2Re}(i - \sqrt{3})$	1
$Re \ll 1, Re/M \ll 1, (\epsilon = 1/Re)$	$\frac{i u}{Re}$	$\frac{u}{2Re}(-0.34 + i2.34)$	$\frac{u}{2Re}(2.34 - i0.34)$	1.18

Table 4.4. Preconditioned Navier-Stokes, $\epsilon = \text{Max}(\epsilon_{inv}, \epsilon_{vis})$

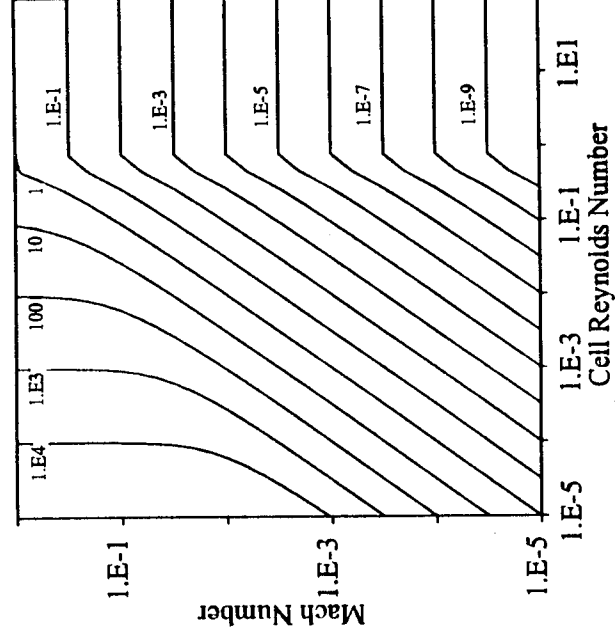


Fig. 4.1. Contours of the preconditioning parameter ϵ as a function of Mach number and cell Reynolds number for the 1-D preconditioned Navier-Stokes equations.

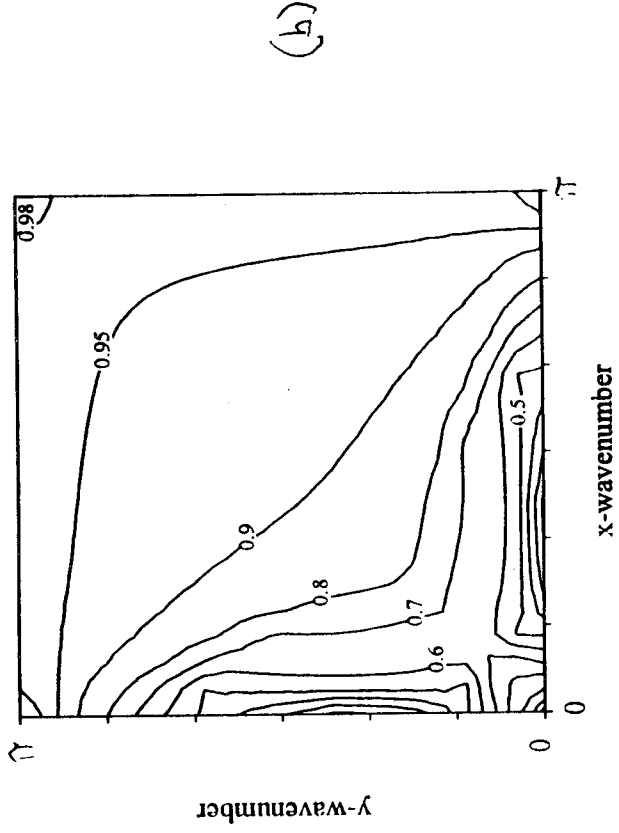
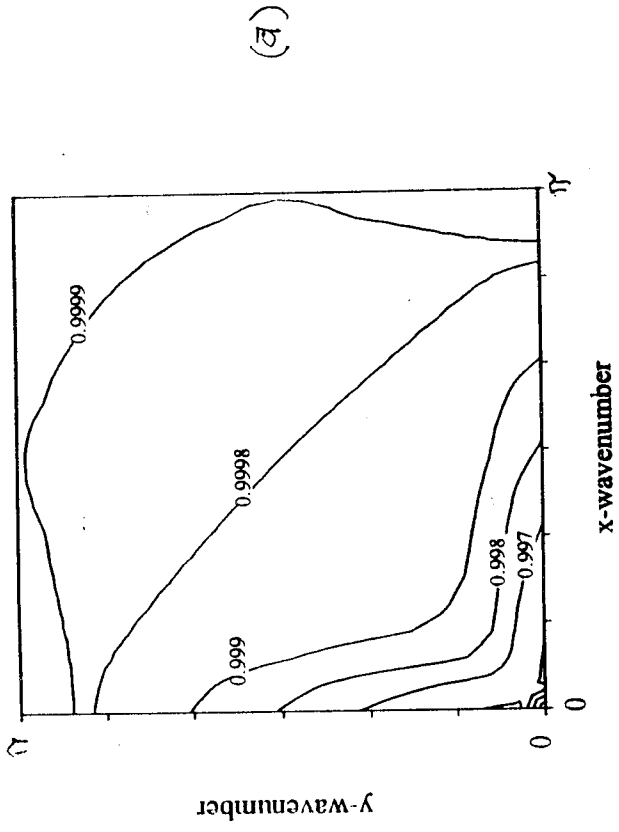


Fig. 4.2. Contours of the maximum eigenvalue of the amplification matrix for the central-difference ADI scheme applied to the 2-D preconditioned Navier-Stokes equations, $M=0.001$, $Re_{\Delta}=0.001$, $v/u=1$, $AR=\Delta x/\Delta y=1$. (a) inviscid preconditioning; (b) viscous preconditioning.

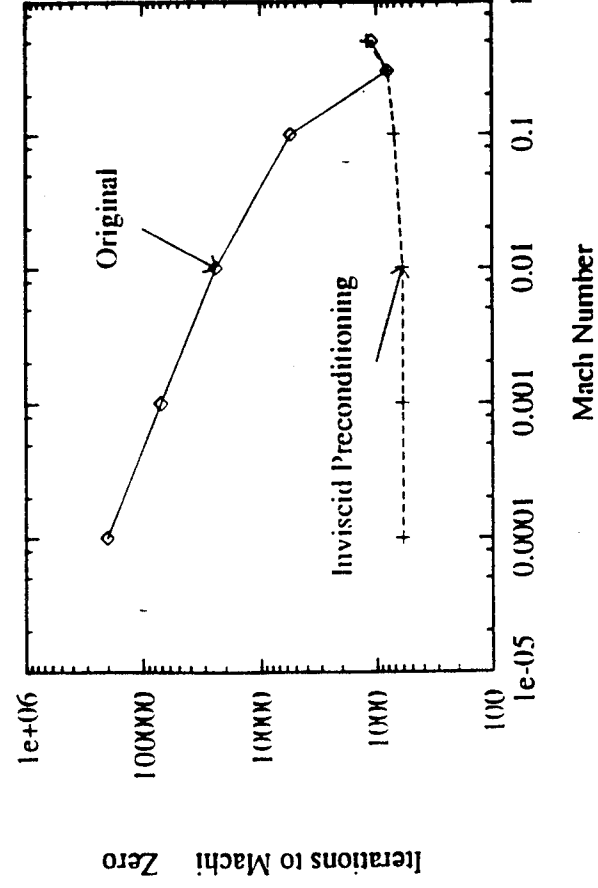


Fig. 4.3. Number of iterations to converge to machine accuracy versus Mach number. Solid line is the standard (no-preconditioning) equations, and the dashed line is with inviscid preconditioning.

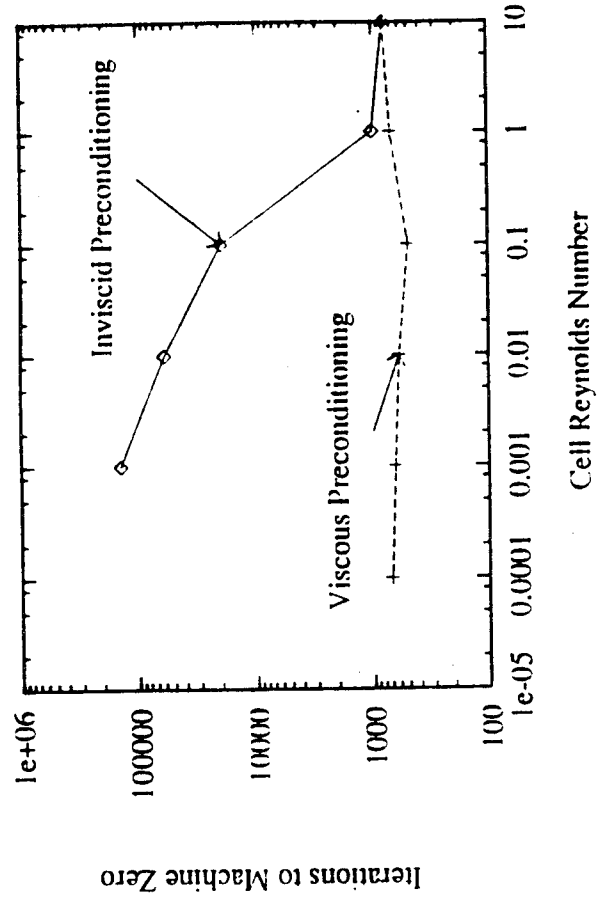


Fig. 4.4. Number of iterations to converge to machine accuracy versus cell Reynolds number. Solid line is with inviscid preconditioning, and the dashed line is with viscous preconditioning.

GRID ASPECT RATIO EFFECTS ON CONVERGENCE OF CENTRAL DIFFERENCED ALGORITHMS

5.1 Time-Step Definition and the Problem at High Aspect Ratios

The convergence characteristics of any numerical Navier-Stokes algorithm are intimately related to the non-dimensional time-steps represented by the various CFL and VNN numbers. The magnitudes of these numbers naturally depend on the specification of the local physical time step (Δt). The proper definition of Δt is therefore crucial to algorithmic performance especially for high aspect ratio problems.

Researchers in *CFD* invariably use local time-stepping to specify a different time step at each grid point. This is generally done by setting the dominant or maximum CFL at any grid location to a specified "optimum" number. Mathematically, this maximum CFL definition is obtained by computing the minimum time step in the two directions,

$$\Delta t = \text{Min} \left[\frac{CFL\Delta x}{\lambda_x}, \frac{CFL\Delta y}{\lambda_y} \right] \quad (5.1)$$

Here, λ_x and λ_y are the maximum eigenvalues in the respective coordinate directions. Even though the ADI algorithm is unconditionally stable, the CFL number is typically taken to be between 1 and 10 for optimum convergence. (The preconditioning matrix ensures that the other inviscid and viscous time scales, such as the fluid velocity and the damping rates, are automatically optimized by the above choice of optimum Δt .)

The time step definition in Eqn. (5.1) illustrates the problem experienced with high aspect ratio grids. When the aspect ratio is much greater than unity, the maximum CFL (CFL_y in this case) is maintained at the optimum value while the CFL in the x direction is much smaller (i.e., $CFL_y = CFL$ and $CFL_x = CFL/AR$). This disparity in the two CFL's results in very slow propagation of waves in the streamwise direction causing poor convergence. The converse situation exists for aspect ratios that are much less than unity. Here, $CFL_x = CFL$ and $CFL_y = AR \cdot CFL$ and errors propagate very slowly in the cross-stream direction. This problem will become more evident in the following sections, where we examine the relationship between this time-scale disparity and the stability characteristics of the algorithm.

To develop an understanding of the manner in which high aspect ratio grids affect convergence and the methods for alleviating these factors, we adopt a two-pronged approach. To obtain a theoretical understanding, we start from stability analyses of the linearized Navier-Stokes equations in vector form. We then test this theoretical insight experimentally by parametric numerical solutions of first simple, and then, more complex and realistic flowfields. To illuminate the separate contributions of the convective and diffusive terms, we

first consider the Euler equations and treat the convective terms alone. We then turn to the complete Navier-Stokes equations where both convective and diffusive effects are present.

5.2 Analysis of Euler Equations

The inviscid version of Eqn. (2.1) (the preconditioned Euler equations) is obtained by setting the diffusive terms to zero. The stability characteristics of this Euler system based on the ADI approximate factorization scheme are given in Figs. 5.1-5.3. In these figures (as well as in the other stability results in this paper), contours of the maximum eigenvalue of the amplification matrix are plotted in (k_x, k_y) wavenumber space. Figure 5.1 shows the results for a grid aspect ratio of unity and two values of CFL, 1 and 10. To conserve space, we take advantage of the symmetry of the stability contours about the line $k_x = \pi/2$ and show results for both values of CFL on one plot. For $CFL = 1$ (the left half of Fig. 5.1), the amplification factor indicates good damping over the whole wavenumber domain while for $CFL = 10$ (the right half of Fig. 5.1), the approximate factorization error causes the amplification factor to approach unity over much of the domain. These results thus demonstrate the well-known ADI result that the optimum CFL lies between 1 and 10.

At an aspect ratio of 100, the results are considerably different as Figs. 5.2 and 5.3 show. Further, we must now distinguish between CFL_x and CFL_y . Figure 5.2 shows stability results for $CFL_x = 0.01$, $CFL_y = 1$ (left half) and $CFL_x = 0.1$, $CFL_y = 10$ (right half). From the traditional maximum CFL time-step definition in Eqn. (5.1), these correspond to specified $CFL_{\max}$ values of 1 and 10 respectively. For both of these cases, the amplification factor is ‘one-dimensional’ and approaches unity along the abscissa of the stability diagram. These results imply that the scheme possesses fairly good damping of cross-stream disturbances but very poor damping of purely streamwise (or longitudinal) disturbances. This is a consequence of the extremely low CFL_x value. Numerical experiments with a preconditioned Euler code confirm that convergence at these values of CFL is seriously affected for high grid aspect ratios and the residual errors suggest that the purely longitudinal modes are the controlling factor for this condition. It is interesting that at the higher CFL condition (right half) in Fig. 5.2, the effect of approximate factorization is not evident as it is in Fig. 5.1.

Stability results for higher values of $CFL_{\max}$, 100 in the left half and 1000 in the right half, are shown in Fig. 5.3. (Here, $CFL_x = 1$, $CFL_y = 100$ on the left half and $CFL_x = 10$, $CFL_y = 1000$ on the right half.) Despite the higher $CFL_{\max}$ value, the amplification factors on the left still show little evidence of the approximate factorization error. The amplification factor is, in fact, well-conditioned not only over the mid-wavenumbers but is now also well-conditioned for purely longitudinal waves. This latter result is a consequence of the fact that CFL_x is unity for this case. In the right half of Fig. 5.3, the effect of approximate factorization finally becomes evident and the amplification factor approaches unity over most of the wavenumber domain. Thus, for an aspect ratio of 100, the optimum $CFL_{\max}$ lies between 100 and 1000. This implies that the optimum CFL_y lies between 100 and 1000, while the optimum CFL_x lies between 1 and 10.

The results in Figs. 5.2 and 5.3 (and numerous other results not shown) suggest that the optimum $CFL_{\max}$ for the ADI scheme scales as the aspect ratio. In other words, a nearly

optimum time step may be obtained for all aspect ratios by specifying the *minimum* CFL (rather than the maximum) to be a fixed value (between 1 and 10) at every grid location. Thus, if the optimum $CFL_{\max}$ is 5 for an aspect ratio of unity, then for AR of ten, it would be 50, while for $AR = 100$, it would be 500 and for $AR = 1 \times 10^6$, the optimum $CFL_{\max}$ would be as high as 5×10^6 . For all of these cases, the optimum time step size may be obtained by choosing CFL_x or the minimum CFL to be 5. In particular, this surprising result indicates that, if the local time step were chosen carefully, the ADI scheme would not suffer serious convergence deterioration at any grid aspect ratio.

Inspection of the amplification factor for a central-differenced ADI scheme applied to a model scalar equation likewise upholds this conclusion. The magnitude of the amplification factor is controlled not by the *absolute* magnitude of the approximate factorization error, but by the *relative* magnitude of this error term with respect to the convective terms. Thus, if a constant value of CFL is to be used for all aspect ratios, that CFL value must be based on the minimum of CFL_x and CFL_y and not on the maximum. Mathematically, this new time-step definition may then be enforced by rewriting the standard time-step definition in Eqn. (5.1) to select the maximum Δt in the two directions rather than the minimum:

$$\Delta t = \text{Max} \left[\frac{CFL\Delta x}{\lambda_x}, \frac{CFL\Delta y}{\lambda_y} \right] \quad (5.2)$$

Since the above definition is based on keeping the minimum CFL at the specified value, we will henceforth refer to it as the min-CFL time-step. In contrast, we will refer to the standard time-step definition in Eqn. (5.1) as the max-CFL time-step. Also note that the min-CFL definition automatically accounts for both longitudinally and transversely stretched grids.

In the computational results presented later, we will see that, when applied by itself, the min-CFL time-step definition suggested by stability theory is generally not sufficient to ensure efficient convergence rates at all aspect ratios. In fact, this alternative is usually unreliable unless it is used in conjunction with properly implemented boundary conditions. We discuss the correct boundary procedures in a later section, but first, we extend these Euler equation stability findings to the Navier-Stokes equations. It is obviously for Navier-Stokes solutions that high aspect ratio grids are most prevalent.

5.3 Analysis of the Navier-Stokes Equations

An important aspect of high aspect ratio Navier-Stokes computations is the significance of viscous preconditioning. High aspect ratio problems are generally dominated by viscous processes in one of the co-ordinate directions (this is one reason why highly stretched grids are employed), even when the streamwise Reynolds numbers are high. The purpose of viscous preconditioning is to optimize the inviscid and viscous modes of the problem simultaneously. For regular-sized grids, the maximum inviscid and viscous time scales are optimized by the viscous preconditioning parameter, ϵ_{vis} , given in Eqn. (4.7). For high aspect ratio grids, the question of which scales must be optimized is not as clear, especially in light of the above findings for the inviscid equations. In this section, we consider the definition of the viscous preconditioning parameter for high aspect ratio situations.

Both scalar and vector analyses of the inviscid equations indicate that the optimum CFL number may be obtained for all grid aspect ratios by fixing the minimum CFL rather than the maximum. A similar analysis of a scalar diffusion equation shows that the optimum VNN is likewise properly determined by the minimum VNN rather than the maximum. Furthermore, for the combined scalar convection-diffusion equation, the optimum time step may be determined by choosing the most restrictive amongst the min-CFL and min-VNN time steps.

If this simple redefinition of the time step carried over directly to the Navier-Stokes equations, the application of Navier-Stokes codes in high aspect ratio grids would be quite straightforward. Unfortunately, the definition of time step for the full Navier-Stokes equations is not so straightforward as the following stability results testify. As a first step, we select the time-step based on the min-CFL. min-VNN definition in analogy with the scalar convection-diffusion case. Accordingly, the viscous preconditioning parameter is chosen by simultaneously optimizing the minimum of (CFL_x, CFL_y) and the minimum of (VNN_x, VNN_y) . This simply involves rewriting Eqn. (4.7) to pick the minimum of the two choices rather than the maximum. Thus, we get:

$$\epsilon_{vis} = \text{Min} \left[\frac{\alpha(\alpha - 1)}{(\alpha - 1 + c^2/u^2)}, \frac{\beta(\beta - 1)}{(\beta - 1 + c^2/v^2)} \right] \quad (5.3)$$

The time-step is then chosen similar to the Euler case as given in Eqn. (5.2). Note that for aspect ratios greater than unity, this choice of time step maintains CFL_x and VNN_x at the optimum value (between 1 and 10) while CFL_y and VNN_y take on much larger values that are proportional to the aspect ratio and aspect ratio squared respectively.

Stability results for the full Navier-Stokes equations are given in Figure 5.4 for $AR = 1000$ using min-CFL, min-VNN definition. Results for $CFL_x = 1$, $VNN_x = 1$ are shown on the left half and for $CFL_x = 10$, $VNN_x = 10$ are shown on the right half. The results on the left half of Fig. 5.4 ($CFL_x = 1$, $VNN_x = 1$, $CFL_y = 1000$ and $VNN_y = 1 \times 10^6$) show good damping over most of the wavenumber domain except for a stiff region at the top of the domain around the $(\pi/2, \pi)$ point where the amplification factor approaches unity. Careful examination reveals that the source of the stiffness in the stability plot is the approximate factorization error term from the cross-product of the streamwise inviscid term and the cross-stream viscous term. The magnitude of this error term may be represented by the product of CFL_x and VNN_y (though this is not precise for the vector system). Because of the large value of VNN_y , this error term is large enough to dominate the other terms in the equation. Exact analysis is difficult for the vector system but the result is an indication of how scalar results do not always carry over to the vector system.

Experiments with a preconditioned Navier-Stokes code for high aspect ratio problems likewise indicate that the min-CFL, min-VNN time step definition does not perform very well. In fact, the convergence rates obtained are quite poor which suggests that the stiffness in the stability result is the controlling factor.

Since the problem associated with the min-CFL, min-VNN definition arises from the approximate factorization product of CFL_x and VNN_y , one way to control the size of the error term is to reduce VNN_y . This may be achieved by optimizing the maximum VNN rather than

the minimum while continuing to specify the minimum CFL number. Such a definition, which we will refer to as min-CFL, max-VNN, would retain the benefits of the min-CFL definition for the ‘inviscid’ modes of the problem while maintaining the conventional restriction on the maximum allowable VNN for the ‘viscous’ modes. Therefore, for aspect ratios greater than unity, the viscous preconditioning parameter, ϵ_{vis} , is chosen to maintain CFL_x and VNN_y at the specified optimum values, while for aspect ratio less than unity, ϵ_{vis} is chosen to maintain CFL_y and VNN_x at the specified values. Mathematically, ϵ_{vis} then depends on two new parameters $\gamma = (CFL \cdot AR)/(VNN \cdot Re_{\Delta y})$ and $\delta = CFL/(VNN \cdot AR \cdot Re_{\Delta x})$ and is given by:

$$\epsilon_{\text{vis}} = \text{Max} \left[\frac{\gamma(\gamma - 1)}{(\gamma - 1 + c^2/u^2)}, \frac{\delta(\delta - 1)}{(\delta - 1 + c^2/v^2)} \right] \quad (5.4)$$

Here, the first choice is selected for aspect ratios greater than unity while the second one is selected for aspect ratios less than unity. The time-step size is selected based on the min-CFL definition and is the same as that given in Eqn. (5.2).

The stability results for the Navier-Stokes equations for AR=1000 using this new min-CFL, max-VNN definition are shown in Fig. 5.5. On the left half, CFL_x = 1, VNN_y = 1 (CFL_y = 1000, VNN_x = 1×10^{-6}) while on the right half, CFL_x = 10, VNN_y = 10 (CFL_y = 1×10^4 , VNN_x = 1×10^{-5}). The amplification factor for the lower CFL and VNN (left half) is seen to be well-conditioned except for the region along the abscissa where it approaches unity. The stiffness in this region is caused by the very small viscous time step in the streamwise direction (VNN_x = 1×10^{-6}) and not the approximate factorization error. While this represents a potential problem for high aspect ratio Navier-Stokes computations, our numerical experiments indicate that it has little effect. In the Results section, we consider several high aspect ratio Navier-Stokes computations. In all cases, very efficient convergence is obtained using the min-CFL, max-VNN time-step. This result suggests that the purely longitudinal waves are important for the ‘inviscid’ modes but not for the ‘viscous’ modes of Navier-Stokes problems. This is not completely surprising since in many Navier-Stokes problems, such as boundary layers and shear layers, the convective terms dominate in the longitudinal direction while the viscous terms dominate in the transverse or cross-stream direction. The choice of the min-CFL, max-VNN time-step essentially equates these two dominant processes and, therefore, provides optimum convergence behaviour.

5.4 Application of Boundary Conditions

Conventional Navier-Stokes codes use a wide variety of boundary condition procedures including extrapolation, reflection and method of characteristics (MOC) procedures. Extrapolation and reflection methods are easier to implement than MOC procedures and, therefore, are quite popular. MOC procedures, though more complicated, are more rigorous since they draw directly from the mathematical character of the unsteady governing equations. Specifically, the characteristics procedure dictates the precise number of conditions to be specified on each boundary, and provides the additional information needed at the boundary through integration of the appropriate characteristic equations.

Past experience with applying various boundary condition procedures has indicated that all of the boundary condition methods (extrapolation and characteristic methods) work equally well when moderate values of CFL are used. Calculations with direct inversion (no approximate factorization), however, show that MOC procedures are much more effective at higher CFL's. Our current analyses of high aspect ratio convergence indicates that, for optimum convergence at all aspect ratios, the $CFL_{\max}$ should scale as the aspect ratio. Therefore, the choice of boundary procedure has a significant impact on the convergence of high aspect ratio computations. In particular, the proper implementation of MOC boundary conditions becomes imperative under such situations.

For approximate factorization schemes, the MOC boundary conditions [18, 19] may be implemented in two ways: 1) boundary conditions are applied after operator factorization (which we will refer to as MOC I), or 2) the boundary conditions are applied before operator factorization (MOC II). The two procedures may be written for the inlet boundary in the traditional delta form as:

MOC I:

$$\left[\frac{\partial \Omega}{\partial Q} + L_s M^{-1} \left(\Gamma + \Delta t \frac{\partial A}{\partial x} \right) \right] \Gamma^{-1} \left[\Gamma + \Delta t \frac{\partial B}{\partial y} \right] \Delta Q = -\Delta t L_s M^{-1} \mathcal{R}^n \quad (5.5)$$

MOC II:

$$\left[S + \Delta t L_s M^{-1} \frac{\partial A}{\partial x} \right] S^{-1} \left[S + \Delta t L_s M^{-1} \frac{\partial B}{\partial y} \right] \Delta Q = -\Delta t L_s M^{-1} \mathcal{R}^n \quad (5.6)$$

where

$$S = \frac{\partial \Omega}{\partial Q} + L_s M^{-1} \Gamma$$

Here, Ω is the boundary condition vector, M^{-1} is the matrix of left eigenvectors and L_s is the selection matrix that chooses the appropriate number of characteristic equations at a given boundary.

We note that MOC II is more accurate, but MOC I has been more commonly employed in *CFD* codes because it is easier to implement. In the Results section, we will examine convergence results using both MOC procedures. These results indicate that for regular-sized grids both methods work equally well but, for high aspect ratio grids, MOC II gives more reliable convergence behavior.

5.5 Multi-Stage Explicit Schemes

As the above results illustrate, the specific manner in which local time stepping is defined has a critical impact on the convergence of implicit schemes on high aspect ratio grids. Here, we digress briefly to discuss time step selection for explicit algorithms. For this purpose, we consider multi-stage Runge-Kutta methods combined with central differencing in space.

Multi-stage schemes for the Navier-Stokes equations take the form,

$$Q_v^k = Q^n + \alpha_k \Delta t \Gamma^{-1} \mathcal{R}^{k-1} \quad k = 1, 2, \dots, N \quad (5.7)$$

where N is the number of steps and $\mathcal{R}$ again denotes the residual. the steady-state portion of Eqn. (2.1). The amplification matrix for this equation is given by

$$G = I + \alpha_N \Delta t Z + \alpha_N \alpha_{N-1} \Delta t^2 Z^2 + \dots + \alpha_N \alpha_{N-1} \dots \alpha_2 \alpha_1 \Delta t^N Z^N \quad (5.8)$$

where Z is the Fourier transform of $\mathcal{R}$. Unlike approximately factored implicit systems, this amplification matrix can be easily diagonalized [20] for the Euler equations. This readily leads to the conclusion that the optimum local time step is given by computing the limiting CFL from the sum of the directional CFL's. Nearly all researchers currently make this time step choice. Unfortunately, both stability analyses and convergence results show that even with this optimum choice of local time stepping, convergence of Runge-Kutta schemes deteriorates rapidly as aspect ratio is increased. The combination of preconditioning, boundary condition implementation and time step selection thus provides no improvements in Runge-Kutta systems. We do, however, note that the arguments are different for the Navier-Stokes equations and that proper viscous preconditioning and time step definition offers some possibility of improvements at high aspect ratio.

5.6 Computational Results

Analysis of algorithm performance for high aspect ratio computations has indicated that the time step definition, viscous preconditioning and boundary condition implementation are the controlling factors. For efficient convergence at all aspect ratios, the local time step must be selected based on the min-CFL definition and the viscous preconditioning matrix must be based on optimizing the maximum VNN simultaneously with the minimum CFL. Furthermore, convergence results point to the importance of proper method of characteristics boundary procedures to ensure reliable convergence at all aspect ratios.

In this section, we consider various test cases—both simple and complex—to demonstrate the performance of the enhanced algorithmic procedures in the presence of high grid aspect ratio regions in the flowfield. First, we consider the simple case of inviscid and viscous flow in a straight duct which allows us to study grid aspect ratio effects parametrically. Then, we look at more realistic flow problems such as multi-species shear layers, high Reynolds number laminar and turbulent boundary layers and turbulent flow through a choked nozzle with cooled walls. All these examples involve local regions where the grid is strongly stretched giving rise to a wide distribution of grid aspect ratios in the flowfield.

5.6.1 Inviscid Flow in Straight Duct

We start by considering the seemingly trivial problem of the solution of the Euler equations for uniform flow in a straight duct. By varying the aspect ratio of the duct and by maintaining a fixed grid size (41 X 41), we change the local grid aspect ratio from unity to 1000. Figure 5.6 shows the number of iterations required to reach machine accuracy for

these computations as a function of the grid aspect ratio. Clearly, it is not necessary to drive convergence to machine error (typically 10^{-16}) for engineering solutions, but it is very useful for diagnosis of algorithm performance. The convergence results in Fig. 5.6 are shown for the standard algorithm (with max-CFL time step and MOC I boundary conditions), the enhanced algorithm (with min-CFL time step and MOC II boundary conditions) as well as an intermediate algorithm (with min-CFL time step but MOC I boundary conditions). All three algorithms give nearly identical convergence rates for an aspect ratio of unity with the residuals reaching machine zero in about 500 iterations. As the aspect ratio is increased, the convergence of the original algorithm suffers quite seriously and, for an $AR = 100$, it requires about 20,000 iterations. This convergence degradation is due to the poor damping of the longitudinal disturbances in the system as can be verified by examining the residual errors of the computation. This behavior is in agreement with the predictions of stability theory.

The results for the intermediate algorithm also show no improvement in the overall convergence for aspect ratios of 10 and 100 even though the time step is selected on the basis of stability theory to ensure optimum convergence. Inspection of the individual convergence rates for this case, however, shows that the min-CFL time step does accelerate convergence dramatically at first, but that after converging several orders of magnitude, the convergence rate changes slope abruptly and flattens considerably. Examination of the residual errors at this stage indicates that the flattening of convergence is caused by accumulation of errors at the inlet and exit boundaries of the system. This suggests that the treatment of boundary conditions (MOC I) is responsible for the behavior. Interestingly, for $AR = 1000$ and higher, the convergence flattening no longer occurs and convergence is attained in about 1000 iterations. Thus, the impact of boundary conditions on convergence is inconsistent; nevertheless, it is evidence that the improved time step helps convergence, but that boundary condition implementation can also be critical.

In contrast, the results for the enhanced algorithm, where the time step is properly defined and the boundary conditions are correctly applied, show that very good convergence is maintained for all aspect ratios examined. Convergence is attained in about 600 iterations in each instance. Thus, the new time step definition indeed provides uniform convergence at all aspect ratios (in agreement with the stability analyses) and the MOC II procedure is more reliable than the MOC I procedure. The associated savings in *CPU* time improves with aspect ratio, reaching nearly two orders of magnitude at $AR = 1000$.

5.6.2 Viscous Flow in Straight Duct

We next present results of a similar parametric study for the solution of the Navier-Stokes equations in Fig. 5.7. Again, we consider flow in a straight duct and vary the local grid aspect ratio by changing the aspect ratio of the flow domain (we vary the radius of the duct keeping the length fixed). We consider aspect ratios ranging from unity up to one million. Aspect ratios of one million are certainly unrealistic for straight duct flows but are often encountered locally in Navier-Stokes computations of turbulent flow. We consider an example of such a flowfield later. In all cases, the velocity profile at the inlet of the duct is taken to be parabolic obviating the need for grid stretching near the wall. The Mach

number of the flow is 1×10^{-5} and the streamwise cell Reynolds number, $Re_{\Delta x}$, is 100. The cross-stream cell Reynolds number, $Re_{\Delta y}$, ranges from 100 down to 10^{-4} depending on the grid aspect ratio.

Convergence results in Fig. 5.7 are shown for the standard algorithm (based on max-CFL, max-VNN), the enhanced algorithm (based on min-CFL, max-VNN) as well as an intermediate algorithm (min-CFL time step but no viscous preconditioning). For all three algorithms, the boundary conditions are enforced using the MOC II procedure which the inviscid results indicated as superior. At an aspect ratio of unity, as in the inviscid case, all three algorithms give excellent convergence behaviour and convergence to machine zero is attained in less than 1000 iterations. At high aspect ratios, however, the convergence results are dramatically different.

The convergence results for the original algorithm are seen to deteriorate considerably as the aspect ratio is increased. At an aspect ratio of 1000, almost 500,000 iterations are necessary for convergence indicating a convergence slowdown of a factor of 500. For higher aspect ratios, the slowdown is even greater and complete convergence cannot be attained within reasonable amounts of computer time.

In contrast, the convergence results for the modified algorithm show uniformly good convergence (about 1000 iterations) for all aspect ratios. The associated savings in computer time can amount to several orders of magnitude; therefore, the modified algorithm can dramatically influence the capability of computing flowfields involving very high aspect ratios.

The results for the intermediate algorithm show the importance of viscous preconditioning for high aspect ratio Navier-Stokes computations. Note that for all cases, $Re_{\Delta x} = 100$ while $Re_{\Delta y} = 100$ for an aspect ratio of unity and is proportionately lower for higher aspect ratios. It is seen that viscous preconditioning becomes an important factor when the aspect ratio is 100 or larger because the corresponding $Re_{\Delta y}$ becomes of order one or less. Under such conditions, the benefits derived by the min-CFL time step are completely negated by the dominant viscous modes of the problem and convergence is as poor as that for the standard algorithm. These results again highlight the fact that the various controlling issues must be addressed in combination to control the convergence behaviour effectively at all aspect ratios.

5.6.3 Flat Plate Boundary Layer

As a further example of Navier-Stokes solutions, we consider both laminar and turbulent flat plate boundary layer flowfields. Because of the steep velocity gradients near the wall, strong grid stretching is required in the direction normal to the wall, particularly at high Reynolds numbers. This wall-stretching, of course, gives rise to high aspect ratio grid cells adjacent to the wall while the grid cells away from the wall are more or less regular-sized. Convergence results using the enhanced algorithm are given in Fig. 5.8 for four different Reynolds numbers using grids with correspondingly different extents of grid-stretching at the wall. The three lower Reynolds numbers (of 4000, 4×10^4 and 4×10^5 are laminar computations and the corresponding maximum grid aspect ratios (located adjacent to the wall) are 10, 30 and 200. The highest Reynolds number (4×10^6) is a turbulent computation

and the maximum grid aspect ratio for this case is 8000 (y^+ is less than 1). In all cases, the flow Mach number is 0.1, and the grid size used is 61 X 61. For the turbulent boundary layer, the algebraic Baldwin-Lomax turbulence model [21] is employed.

The convergence rates in Fig. 5.8 using the enhanced algorithm are extremely good for all four Reynolds numbers. In particular, despite the different aspect ratios of the grid cells for each of the cases, the convergence rates are almost identical reaching machine zero is about 800 iterations. For comparison, the convergence of one of the laminar cases ($Re = 4 \times 10^5$) using the standard algorithm (based on max-CFL, max-VNN) is also shown in Fig. 5.8. The residuals are seen to drop just over an order of magnitude for every 1000 iterations and it takes about 15,000 iterations to reach machine zero (not shown in Fig. 5.8). Thus, the enhanced algorithm speeds up convergence by a factor of about 20 for this case. As the *CPU* savings typically increases as the grid is further refined, the savings for turbulent boundary layer computations are significantly greater.

The velocity profiles obtained for the $Re = 4 \times 10^5$ laminar calculation and the $Re = 4 \times 10^6$ turbulent calculation are shown in Figs. 5.9 and 5.10. The laminar solution (Fig. 5.9) is compared with the Blasius profile, while the turbulent profile (Fig. 5.10) is plotted against the law of the wall and the log law profiles. In both instances, there is good agreement between the numerical predictions and theory, thereby confirming that the improved algorithm converges to the correct solution.

5.6.4 Laminar Backward-Facing Step

The geometry used for the backward facing step corresponds to the one studied experimentally by Armaly *et al.* [22]. The length and width of the duct are 10 cm and 1 cm respectively, and the step height is 0.5 cm. A uniform grid of 101 X 61, which gives a grid aspect ratio of 6, is used. The predicted recirculation lengths from both codes are compared with the experimental data in the 100 to 400 Reynolds number range (at higher Re , the experiments demonstrated three-dimensionality).

Before looking at the flowfield solutions, we present convergence results. Figure 5.11 shows the convergence for $Re = 100$ using the implicit preconditioned code ($CFL = 4$ and $VNN = 4$). Curve I shows the convergence with the conventional time-step definition—i.e., based on maximum eigenvalue (but with preconditioning), while in Curve II, the time-step definition is based on Eqn. (5.2). The new definition yields superior convergence with about 8 orders of magnitude reduction in the residuals in 1000 steps as opposed to 4 orders in 4000 steps, an improvement of a factor of about 8.

Figure 5.12 shows the velocity contours for the four cases computed. The plots show only the first 6 cm of the flow domain since beyond this point the flow is essentially fully developed. Only the implicit code solutions are shown because the two codes give virtually identical results. The recirculation zone lengths are also marked alongside the plots. At $Re = 400$, a second recirculation zone develops on the opposite wall downstream of the primary recirculation zone. This secondary zone has been observed experimentally and is evident in the velocity measurements at a Re of 389 [22].

In Fig. 5.13, the predicted primary recirculation zone lengths, normalized by the step height (S), are compared against the measured values. The results from the implicit ADI

code described here as well as using an explicit Runge-Kutta algorithm are shown. The experimental values are obtained from the curve-fitted plot of the experimental data in Ref. [22]. As can be seen, the agreement between predictions and measurements is excellent. The slight discrepancy at $Re = 400$ arises because of the three-dimensional effects in the flowfield [22].

5.6.5 Converging-Diverging Nozzle

The final case that we examine is high Reynolds number turbulent flow through a converging-diverging nozzle with cooled walls. The grid geometry and flowfield solution are shown in Fig. 5.14. The incoming gas is at a stagnation temperature of 3500 K while the wall is maintained at 700 K. Of particular interest, for this case, is the accurate prediction of the heat flux to the nozzle wall. Because of the extremely thin boundary layer in the throat region, very strong grid stretching is necessary to maintain a minimum y^+ of about 1 along the wall.

The convergence results are shown in Fig. 5.15. With the enhanced algorithm, rapid convergence is obtained with machine zero being reached in less than 2000 iterations. With the standard algorithm, the convergence is fairly good for about four orders of magnitude reduction in the residual beyond which it becomes very poor. Examination of the residuals indicates that the initial portion corresponds to solution convergence in the central portion of the flowfield where the grids are more or less regular, while the latter portion is due to poor convergence in the high aspect ratio near-wall region. This is also readily apparent by observing the temporal convergence of the wall heat flux shown in Fig. 5.16 for both the standard and enhanced algorithms.

The results in Fig. 5.16 for the standard algorithm without aspect ratio control indicate that after a three order-of-magnitude reduction in the residual (after 2000 iterations), the wall heat flux is far from converged. Converging the flowfield an additional order of magnitude (after 4000 iterations) causes the wall heat flux to change by almost a factor of two. It requires roughly 20,000 iterations (about six orders of magnitude convergence) for the heat flux to be approximately the same as the fully converged result. On the other hand, with the enhanced algorithm, the wall heat flux converges at the same rate as the overall flowfield and a three order-of-magnitude reduction in residual (less than 400 iterations) is sufficient to give wall heat flux to engineering accuracy. Thus, the convergence enhancement methods discussed herein can significantly impact realistic flow computations.

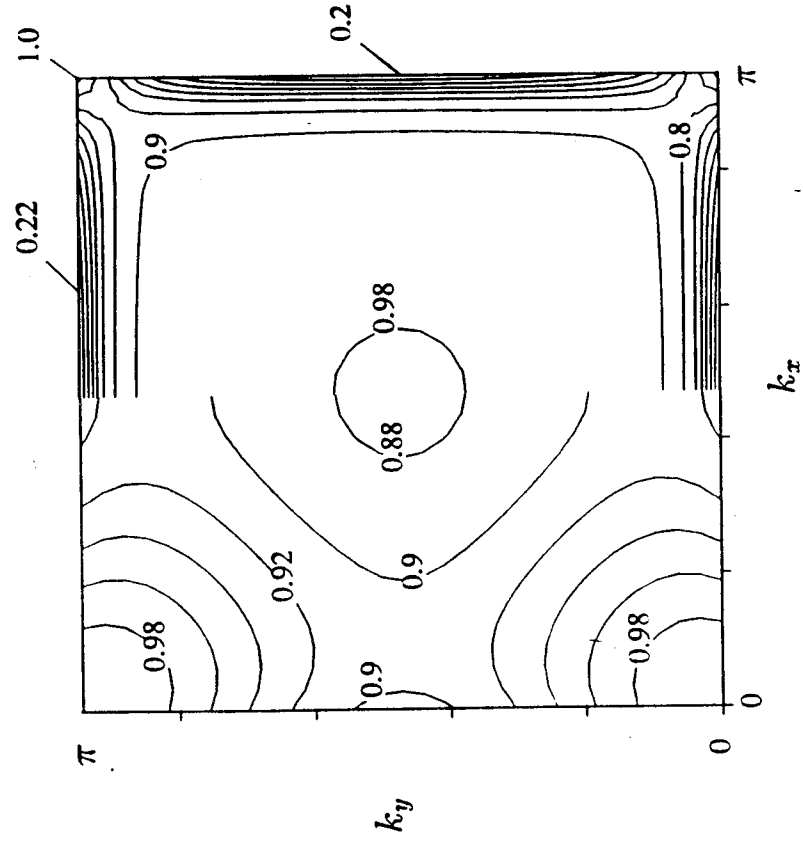


Fig. 5.1. Euler stability for the 2-D central-difference ADI scheme (AR=1). CFL=1 (left side), CFL=10 (right side).

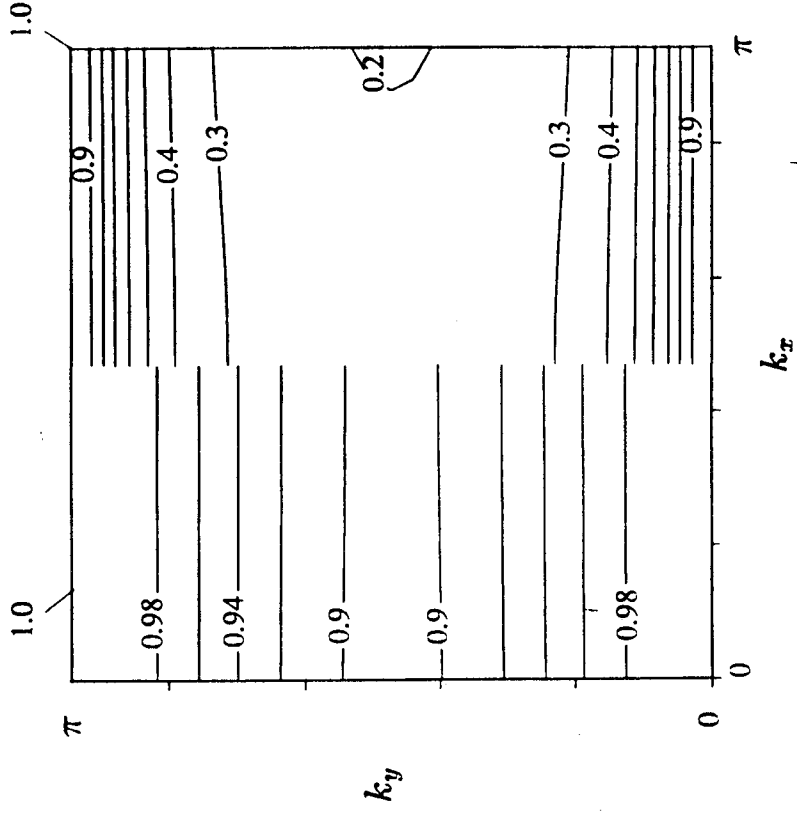


Fig. 5.2. Euler stability for the 2-D central-difference ADI scheme ($AR=100$). Left side - $CFL_x=0.01$, $CFL_y=1$; right side - $CFL_x=0.1$, $CFL_y=10$.

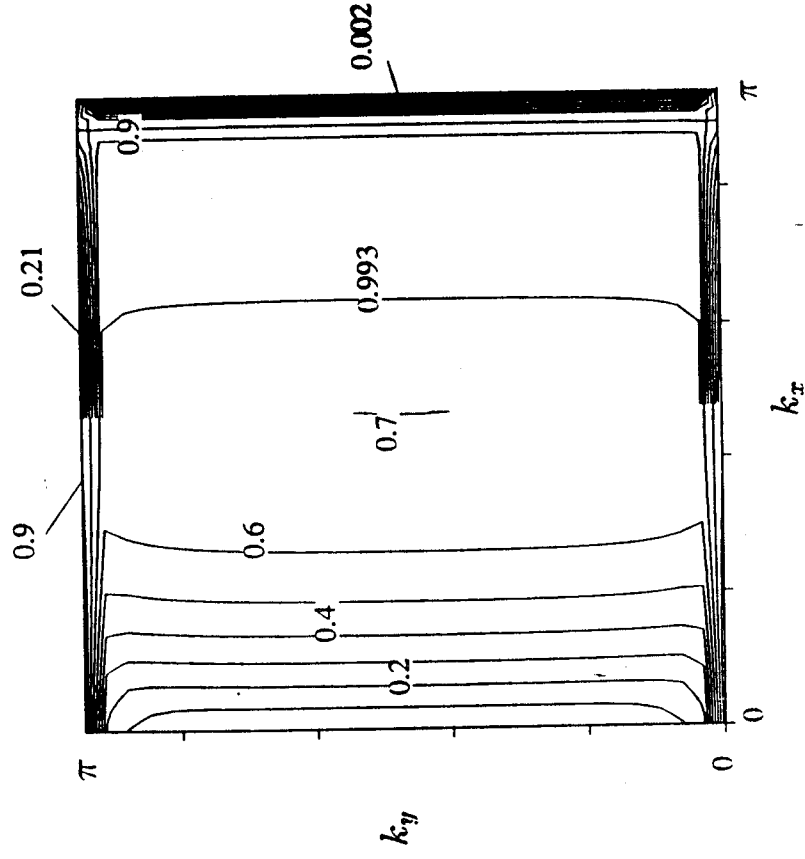


Fig. 5.3. Euler stability for the 2-D central-difference ADI scheme (AR=100). Left side - CFL_x=1, CFL_y=100; right side - CFL_x=10, CFL_y=1000.

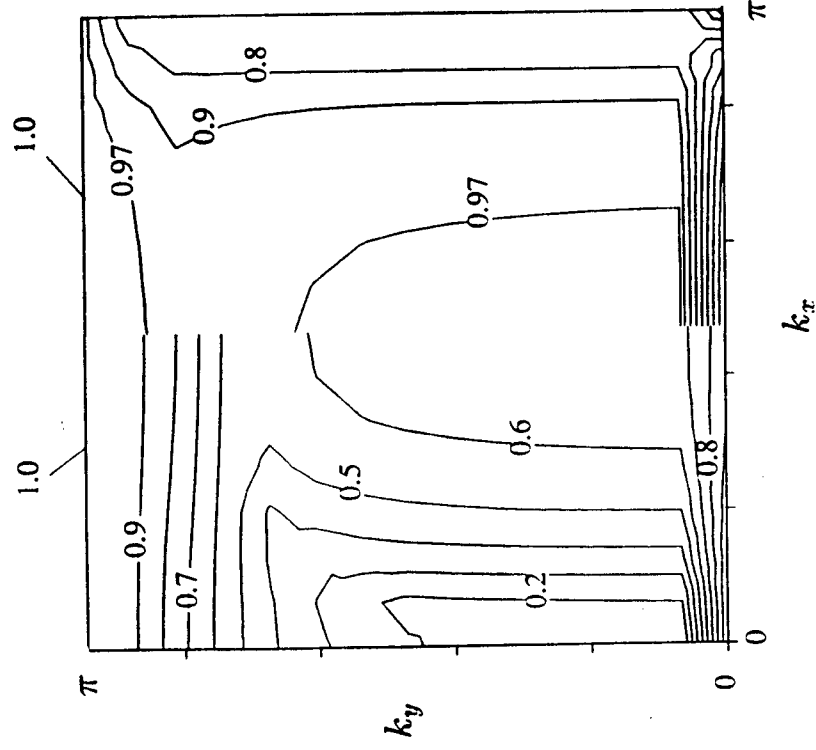


Fig. 5.4. Navier-Stokes stability for the 2-D central-difference ADI scheme ($AR=1000$). Left side - $CFL_x=1$, $VNN_x=1$; right side - $CFL_x=10$, $VNN_x=10$.

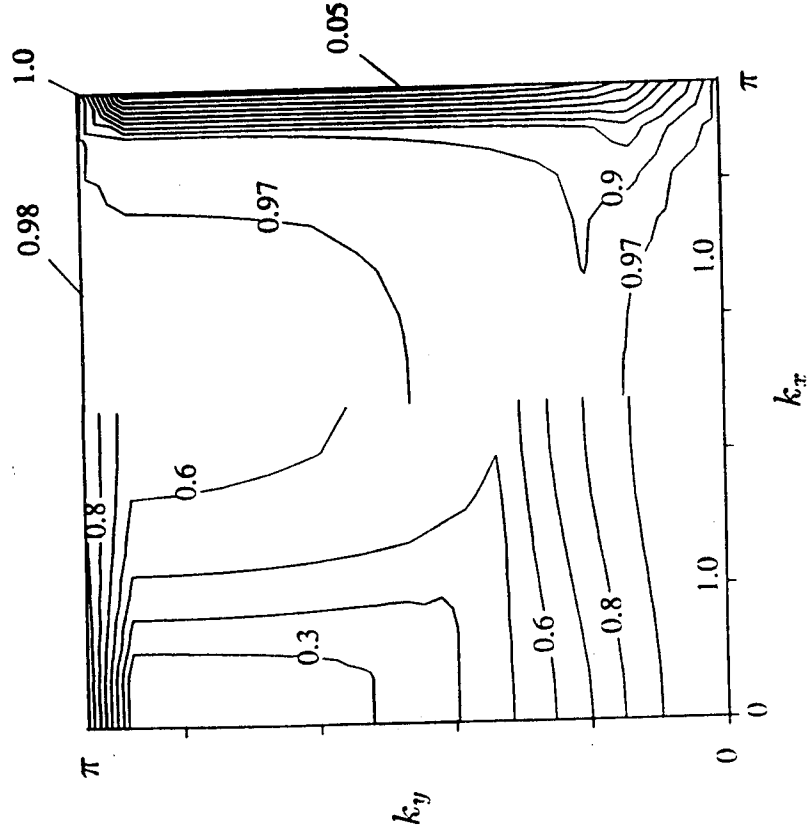


Fig. 5.5. Navier-Stokes stability for the 2-D central-difference ADI scheme ($AR=1000$). Left side - $CFL_x=1$, $VNN_y=1$; right side - $CFL_x=10$, $VNN_y=10$.

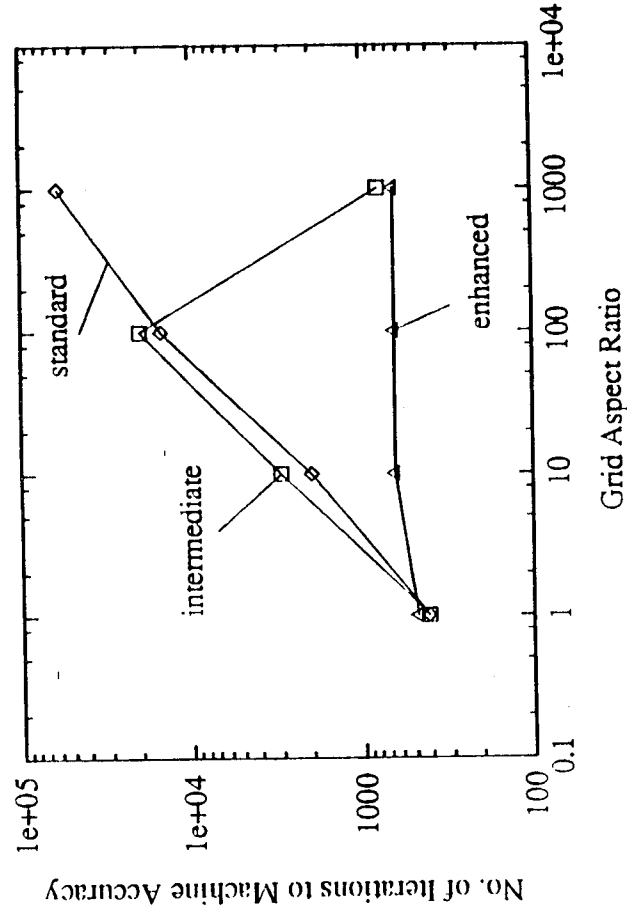


Fig. 5.6. Convergence to machine accuracy for inviscid duct flow with aspect ratio (AR) as a parameter. Standard = MOC I, max-CFL; intermediate = MOC I, min-CFL; enhanced = MOC II, min-CFL.

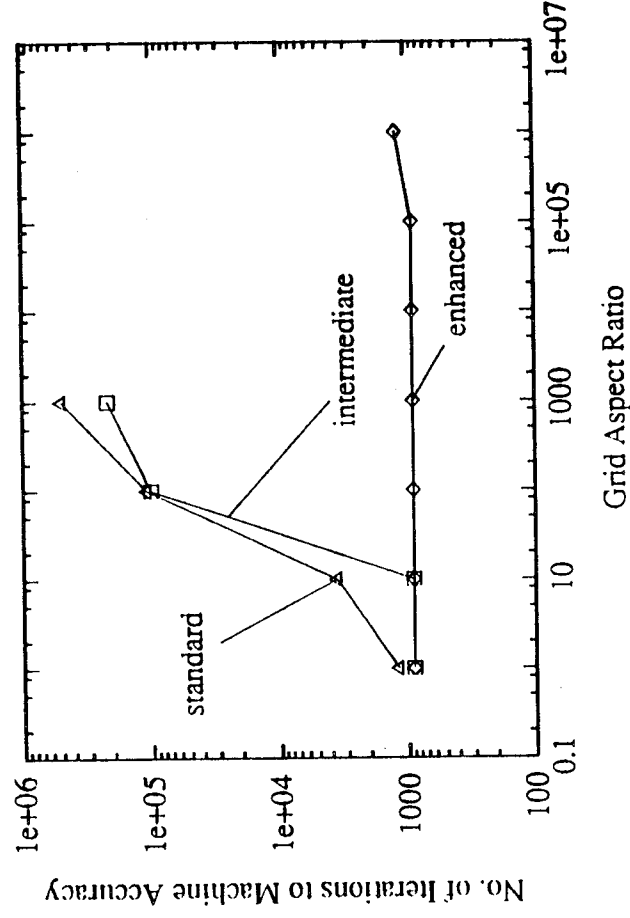


Fig. 5.7. Convergence to machine accuracy for fully-developed viscous duct flow with aspect ratio (AR) as a parameter and $Re_{\Delta x}=100$. Standard = max-CFL, max-VNN; intermediate = min-CFL, no viscous preconditioning; enhanced = min-CFL, max-VNN.

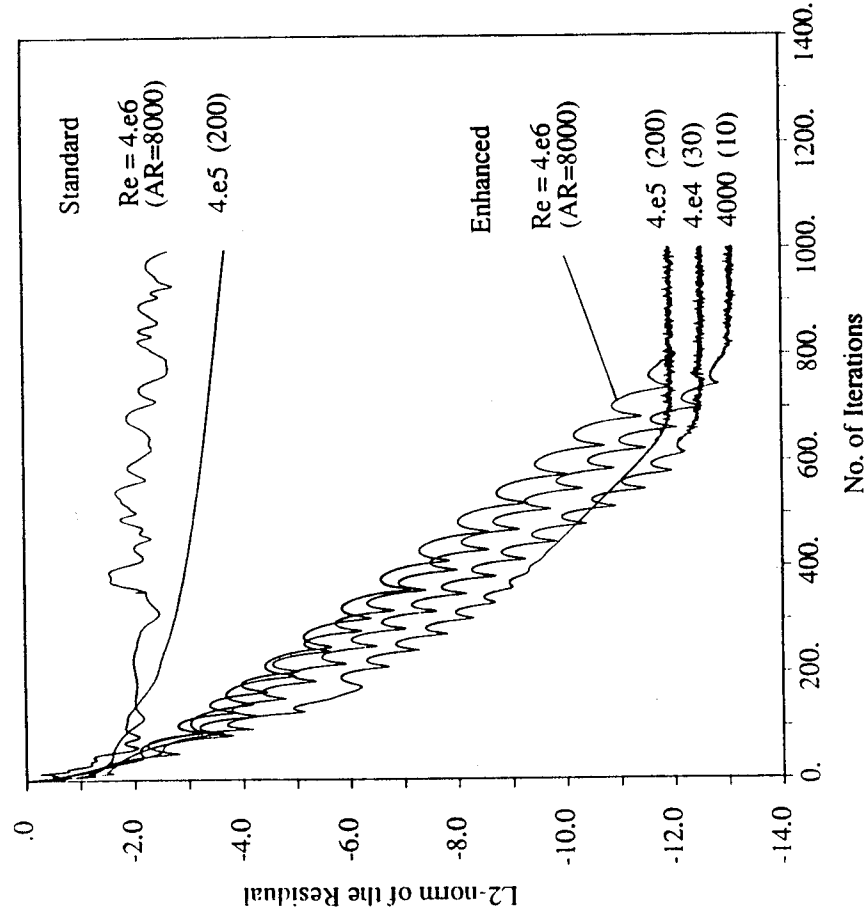


Fig. 5.8. Convergence for flat plate boundary layer for standard and enhanced algorithms.

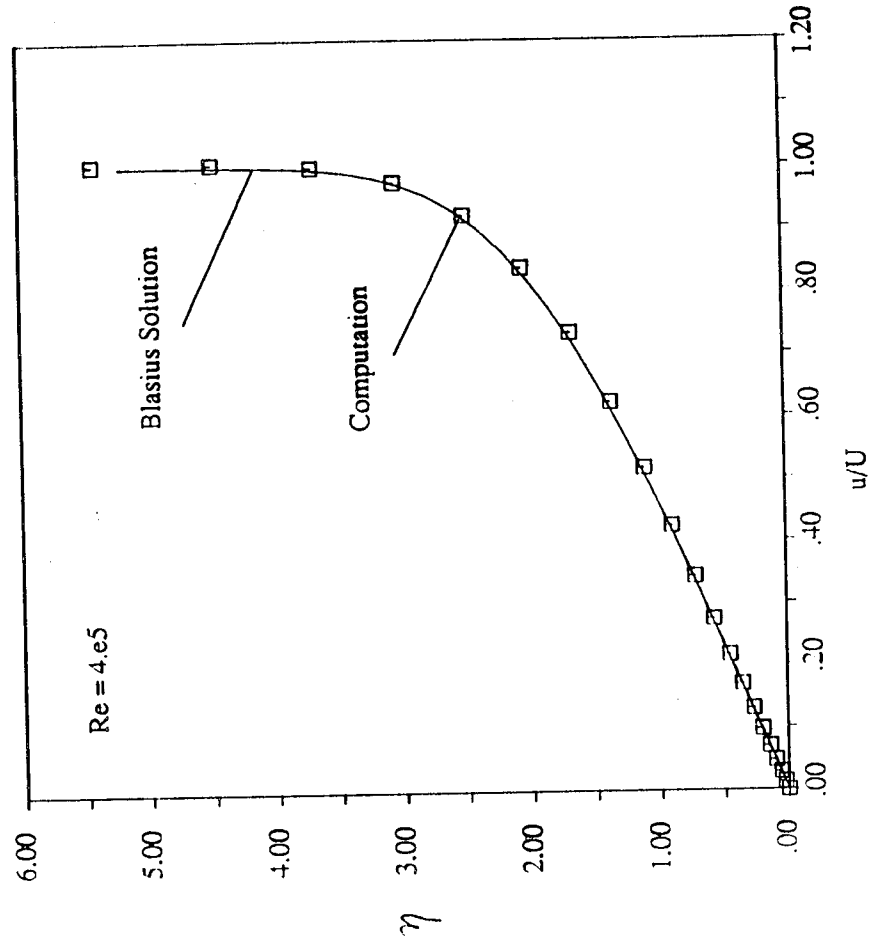


Fig. 5.9. Laminar boundary layer solution.

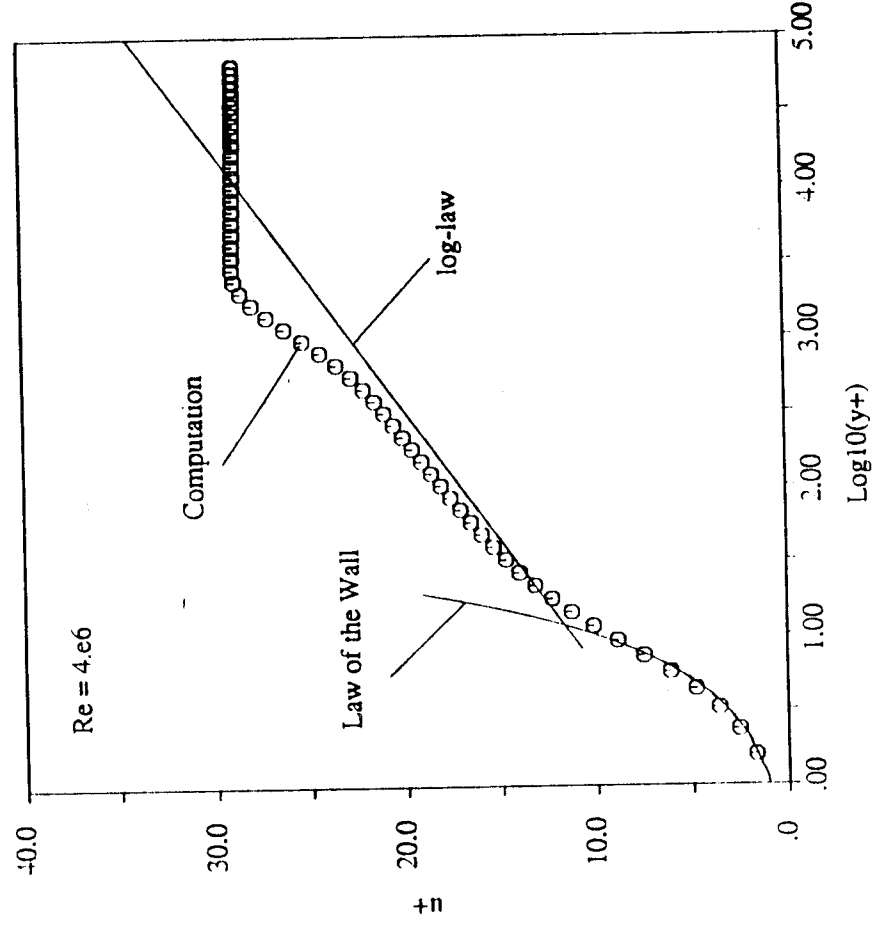


Fig. 5.10. Turbulent boundary layer solution.

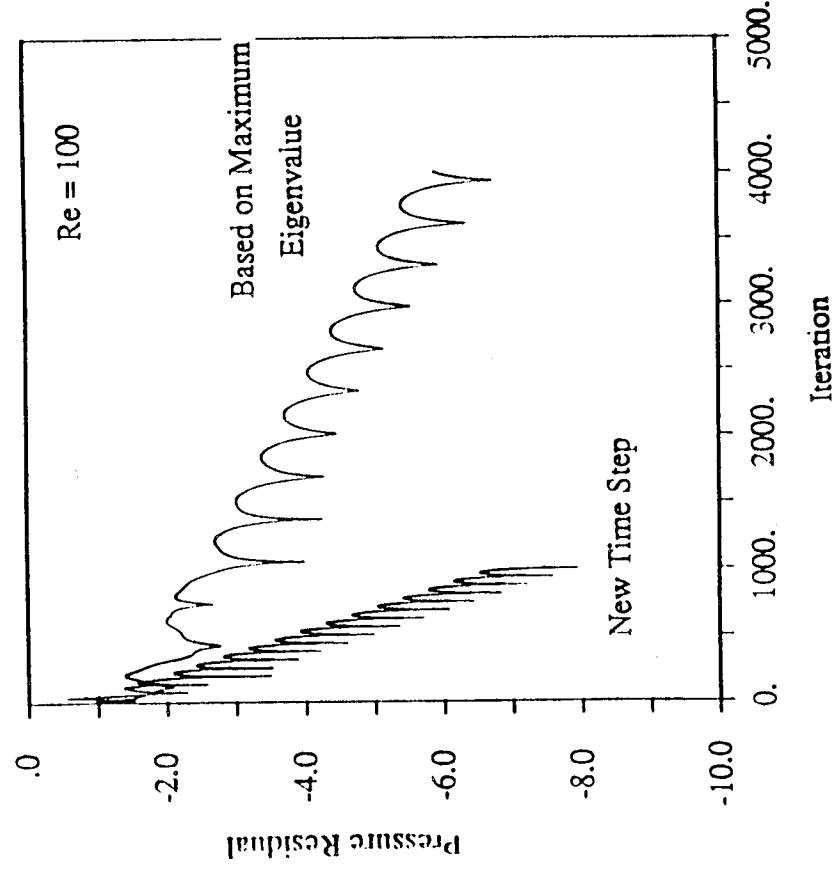


Fig. 5.11. Convergence histories for the 2-D backward-facing step problem using the ADI algorithm. $Re=100$. CFL=4, VNN=4.

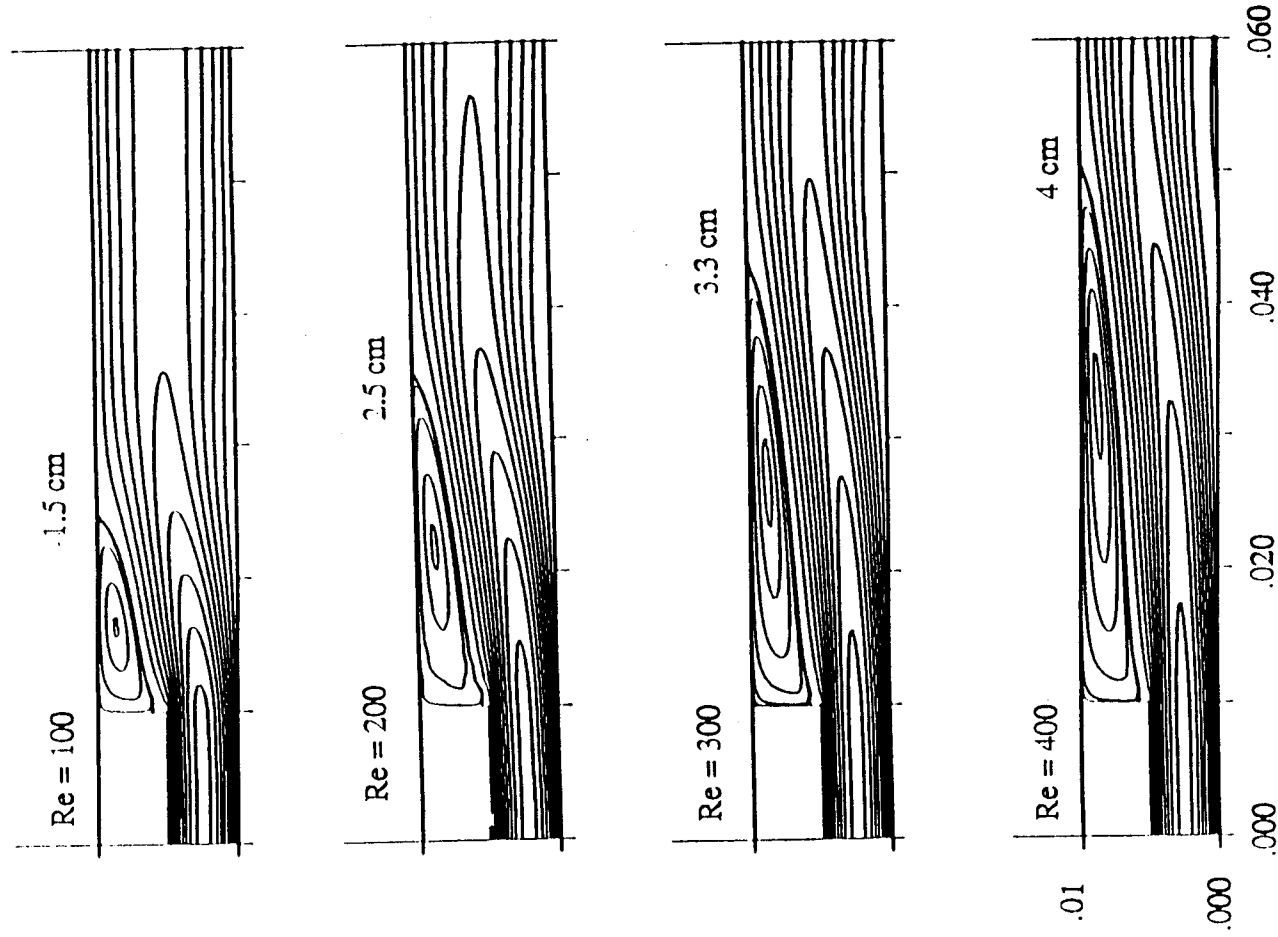


Fig. 5.12. Velocity contours for 2-D backward-facing step problem with Reynolds numbers 100, 200, 300, 400.

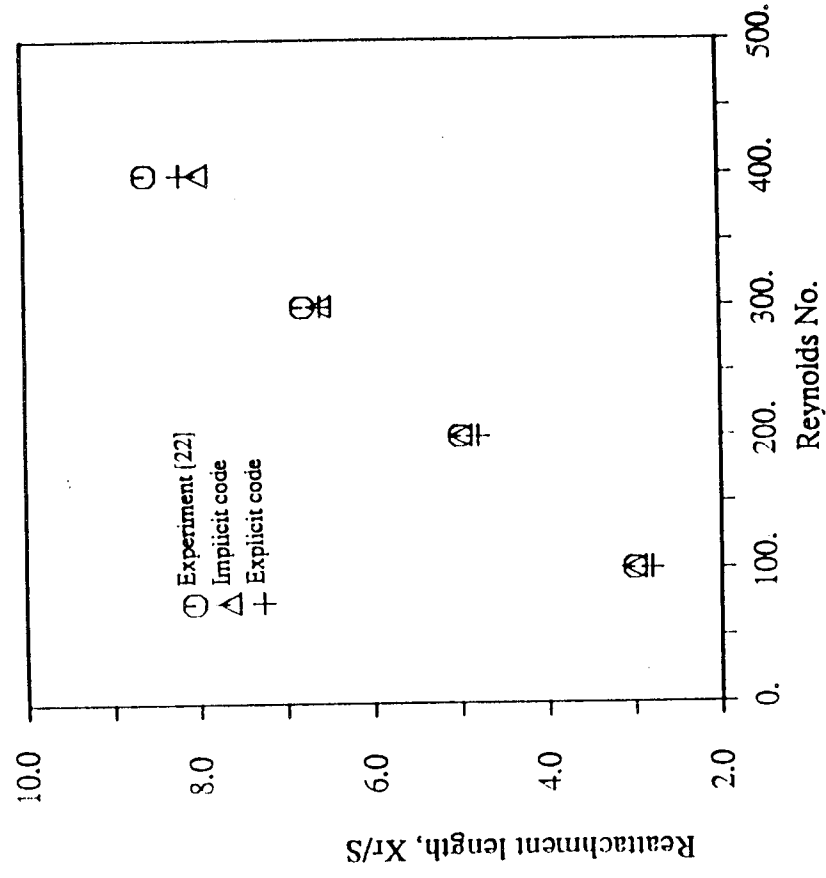


Fig. 5.13. Predicted primary recirculation zone lengths compared with experimental values in Ref. [22].

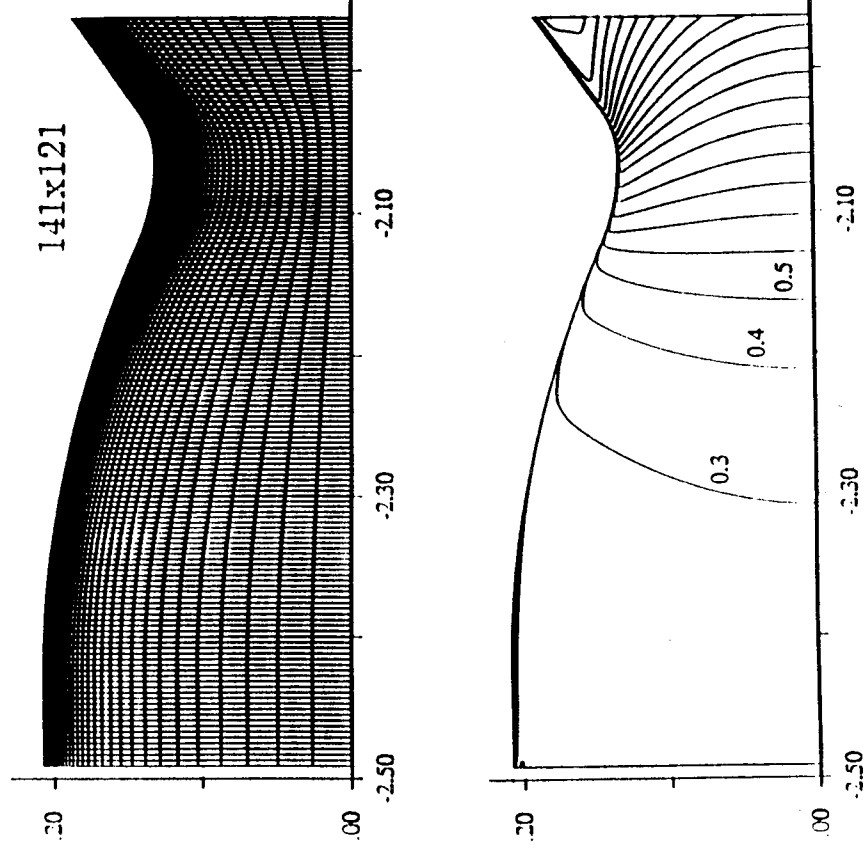


Fig. 5.14. Stretched grid and converged Mach contours for turbulent nozzle computation.

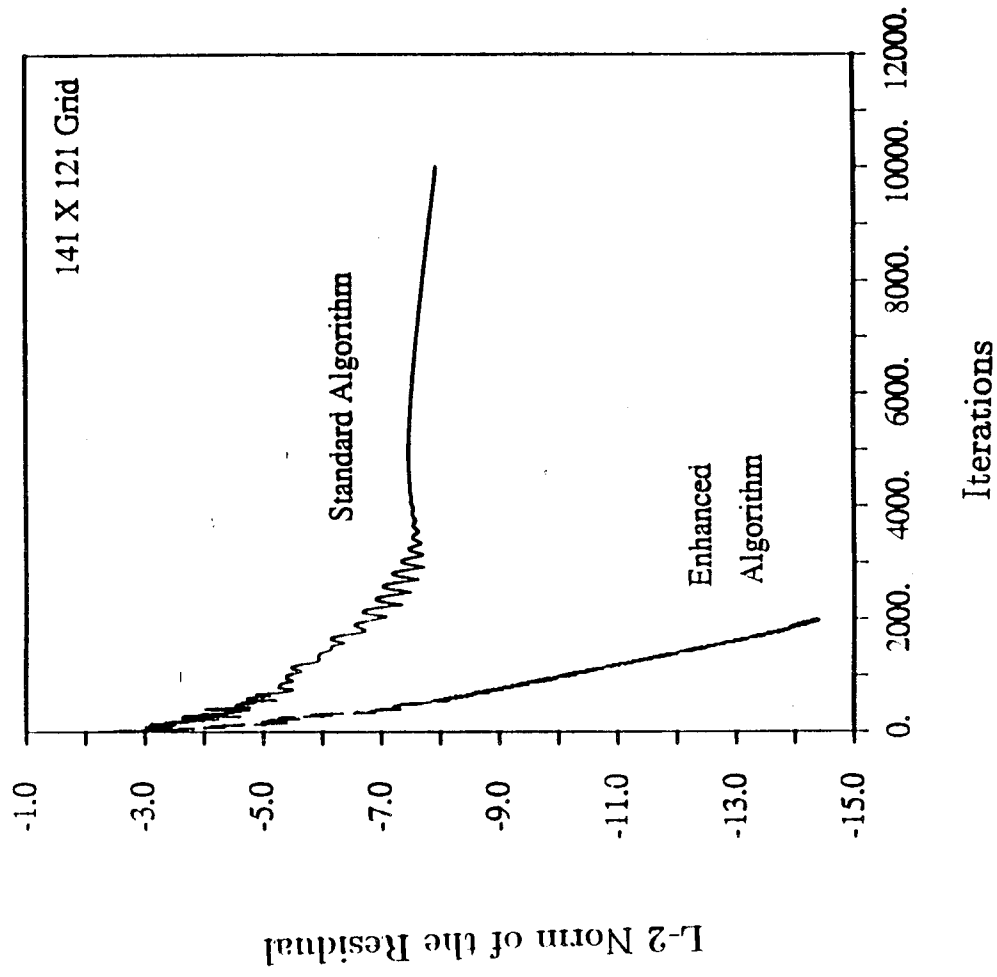


Fig. 5.15. Convergence histories for the turbulent nozzle computation using the standard algorithm and the enhanced algorithm.

Standard Algorithm

Modified Algorithm

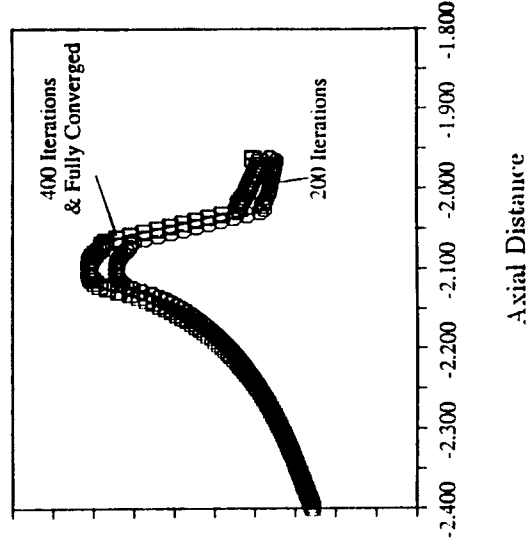
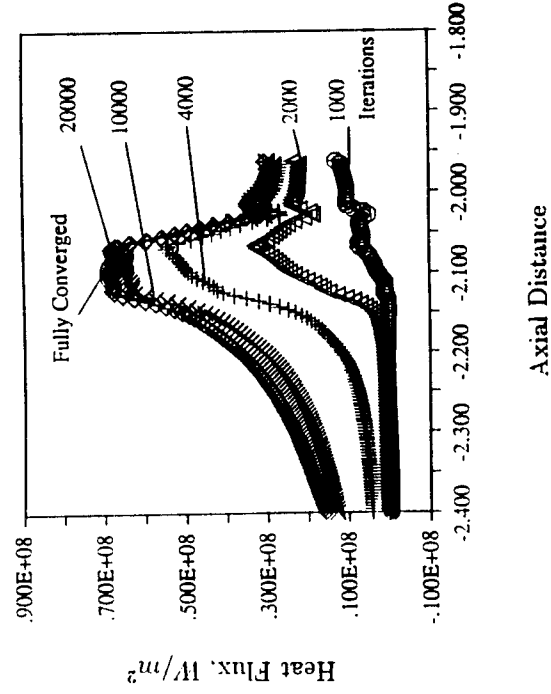


Fig. 5.16. Temporal convergence of the heat flux along the nozzle wall for both the standard algorithm and the enhanced algorithm.

PRECONDITIONING FOR TURBULENT AND REACTING FLOWS

6.1 Governing Equations

The multi-component Navier-stokes equations coupled to the $k - \epsilon$ turbulence model may be written in their traditional conservative form as follows:

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = H + \frac{\partial}{\partial x} R_{xx} \frac{\partial Q_v}{\partial x} + \frac{\partial}{\partial x} R_{xy} \frac{\partial}{\partial y} Q_v + \frac{\partial}{\partial x} R_{yx} \frac{\partial}{\partial x} Q_v + \frac{\partial}{\partial y} R_{yy} \frac{\partial}{\partial y} Q_v \quad (6.1)$$

where the traditional vector of dependent variables Q has been used. The flux vectors are defined as:

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \\ \rho k \\ \rho \epsilon \\ \rho Y_i \end{pmatrix} \quad E = \begin{pmatrix} \rho u \\ (\rho u^2 + p) \\ \rho uv \\ (e + p)u \\ \rho uk \\ \rho u \epsilon \\ \rho u Y_i \end{pmatrix}, \quad F = \begin{pmatrix} \rho v \\ \rho uv \\ (\rho v^2 + p) \\ (e + p)v \\ \rho vk \\ \rho v \epsilon \\ \rho v Y_i \end{pmatrix}$$

Here, $i = 1, 2, \dots, N - 1$, where N is the total number of chemical species. The viscous matrices R_{xx} , R_{xy} , etc. are similar to those given in Eqn. (2.1) with additional entries for the species and turbulence transport equations.

The combustion source term in the species equations, $\dot{w}_i$, is obtained by summing the species generation contributions from the various elemental reactions. This is written as:

$$\dot{w}_i = M_i \sum_{k=1}^{NR} (\dot{c}_i)_k$$

where NR is the total number of reactions and $(\dot{c}_i)_k$ is the rate of production of moles of species i by reaction k ,

$$(\dot{c}_i)_k = (\nu_{ik}'' - \nu_{ik}') (k_{fk} \prod_{m=1}^N C_m^{\nu_{mk}'} - k_{bk} \prod_{m=1}^N C_m^{\nu_{mk}'})$$

where the k^{th} reaction is written as:

$$\sum_{i=1}^N \nu'_{ik} M_i = \sum_{i=1}^N \nu''_{ik} M_i$$

Here, ν'_{ik} and ν''_{ik} are the stoichiometric coefficients for the i^{th} species, k_{fk} and k_{bk} are the forward and backward reaction rates and M_i is the molecular weight of the i^{th} species. C_i is the molar concentration of the species and is given by $C_i = \frac{\rho Y_i}{M_i}$.

The source terms in the k and ϵ transport equation correspond to the standard $k - \epsilon$ model [23] with additional low Reynolds number terms introduced for near-wall effects [24]. These may be written as:

$$H_k = \mu_t S - \frac{2}{3} \rho k \mathcal{D} - \rho \epsilon - \frac{2\mu k}{n^2}$$

$$H_\epsilon = C_1 \mu_t \frac{\epsilon}{k} S - \frac{2}{3} C_1 \rho \epsilon \mathcal{D} - C_2 f_2 \frac{\rho \epsilon^2}{k} - \frac{2 f_1 \mu \epsilon}{n^2}$$

where $\mu_t = C_\mu f_\mu \frac{\rho k^2}{\epsilon}$ and S and $\mathcal{D}$ are given by:

$$S = \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \frac{\partial u_i}{\partial x_j}$$

$$\mathcal{D} = \frac{\partial u_i}{\partial x_i}.$$

6.2 Derivation of the Preconditioned System

The derivation of the preconditioned system for the multi-component Favre-averaged Navier-Stokes equations follows the procedure for the standard Navier-Stokes equations outlined in Ref. [7]. There are, however, several important issues to be considered when altering the time-derivative, some of which are unique to the multi-component equations. Firstly, the preconditioned time-derivative should remain well-posed with respect to the diffusion terms. To ensure this, we transform the equations to the non-conservative form. (van Leer *et al.* have suggested an alternative symmetric form of the equations for the same purpose [10]). Other specific issues concern the choice of independent variables, eigenvalues and eigenvectors which we will discuss later. We may write the non-conservative equations in vector form by transforming the standard conservative equations:

$$K_3 \left[\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = H + L(Q_o) \right] \quad (6.2)$$

The vector $Q = (\rho, \rho u, \rho v, e, \rho Y_i)^T$, the standard conservative set of dependent variables. The viscous operator L is given as:

$$L = \frac{\partial}{\partial x} R_{xx} \frac{\partial}{\partial x} + \frac{\partial}{\partial x} R_{xy} \frac{\partial}{\partial y} + \frac{\partial}{\partial y} R_{yx} \frac{\partial}{\partial x} + \frac{\partial}{\partial y} R_{yy} \frac{\partial}{\partial y}$$

The matrix K_3 contains all the row transformations required to obtain the non-conservative form and is given as:

$$K_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -u & 1 & 0 & 0 & 0 & 0 & 0 \\ -v & 0 & 1 & 0 & 0 & 0 & 0 \\ -h + (u^2 + v^2)/2 & -u & -v & 1 & -\frac{5}{3} & 0 & 0 \\ -k & 0 & 0 & 0 & 1 & 0 & 0 \\ -\epsilon & 0 & 0 & 0 & 0 & 1 & 0 \\ -Y_i & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The first step in deriving the preconditioning matrix is to transform the time-derivative terms in Eqn. (6.2) to a new set of primary dependent variables, $Q_v = (p_t, u, v, T, k, \epsilon, Y_i)^T$, where $p_t = p + \frac{2}{3}\rho k$. The choice of pressure as a dependent variable follows the development for laminar flows as outlined in Ref. [7]. It allows the propagation of the acoustic waves in the system to be controlled by the preconditioning procedure. Here, however, we choose a modified ‘turbulent’ pressure as the dependent variable. This is because the selection of the normal gas pressure as the dependent variable results in complex eigenvalues for high values of turbulence when preconditioning is employed. This may sometimes lead to ill-posedness and erratic convergence behaviour. In contrast, the choice of the turbulent pressure ensures that the eigenvalues of the preconditioned turbulent system are the same as those obtained in the laminar case (Eqn. 2.4). Note that this modified pressure reduces to the standard gas pressure in the absence of turbulence and is therefore consistent with the laminar equations.

The choice of temperature and species mass fraction are primarily a matter of convenience. Other researchers have preferred to use enthalpy and ρY_i [25]; however, T and Y_i simplify the computation of Jacobians and transformation matrices, thus by-passing some of the algebra encountered conventionally in reacting flow calculations [26]. Accordingly, the governing equations may be written as:

$$K \frac{\partial Q_v}{\partial t} + K_3 \left[\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} \right] = H + L(Q_v) \quad (6.3)$$

where

$$K = K_3 \left(\frac{\partial Q}{\partial Q_v} \right) = \begin{pmatrix} 1/R'T & 0 & 0 & -\frac{\rho}{T} \frac{R'}{R} & -\frac{2}{3} \frac{\rho}{R'T} & 0 & -\rho \frac{R_i - R_N}{R} \\ 0 & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & \rho C_p & 0 & 0 & \rho(h_i - h_N) \\ 0 & 0 & 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho \delta_{ij} \end{pmatrix}$$

Note that Eqn. (6.3) still represents the full form of the unsteady equations in Eqn. (6.1). The eigenvalues of the system are $u + c_t$, $u - c_t$, u , u , etc. Note that c_t is the modified frozen sound speed given by $c_t = \sqrt{\gamma(R + \frac{2}{3}k/T)T} = \sqrt{\gamma R'T}$. The coefficient of the pressure time-derivative ($1/RT$) in the continuity equation establishes the speed of propagation of acoustic waves in the system. To control the acoustic speeds at low Mach numbers and low Reynolds numbers, we add a scaling factor, ϵ , to this coefficient. The physical acoustic speeds are then replaced by pseudo-acoustic speeds that will allow proper conditioning of the characteristic speeds of the system at all Mach numbers and Reynolds numbers. The preconditioning matrix $\tilde{\Gamma}$ (which replaces K in Eqn. 4.2) is written as:

$$\tilde{\Gamma} = \begin{pmatrix} 1/\epsilon c_t^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & \rho C_p & 0 & 0 & \rho(h_i - h_N) \\ 0 & 0 & 0 & 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho \delta_{ij} \end{pmatrix}$$

the preconditioning parameter ϵ is defined as given earlier in Eqn. (4.5).

In deriving Γ_v from K , the off-diagonal terms in the first row (i.e., the continuity equation) are dropped. This step is not essential and does not affect the eigenvalues of the system; however, it does affect the selection of eigenvectors and appears to be somewhat beneficial [12]. Furthermore, it allows us to interpret the continuity equation as an equation for updating the pressure field. This, in turn, renders the preconditioned system similar to the artificial compressibility method for incompressible flows [27] as well as to the pressure-based methods referred to earlier [12]. We also note that, for the multi-component Navier-Stokes equations, special care needs to be exercised in selecting the preconditioning matrix. For instance, an earlier choice of preconditioning matrix, wherein the $\rho(h_i - h_N)$ term was dropped, did not possess a complete set of linearly independent eigenvectors, which clearly renders the system ill-posed.

The final step in deriving the preconditioning matrix is to transform the preconditioned equations to conservative form by premultiplying the non-conservative equations by K_3^{-1} . The final preconditioning matrix is given by $K_3^{-1}\tilde{\Gamma} = \Gamma_p$, and the equations become:

$$\Gamma_p \frac{\partial Q_v}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = H + L(Q_v) \quad (6.4)$$

where

$$\Gamma_p = \begin{pmatrix} 1/\epsilon t_i^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ u/\epsilon c_t^2 & \rho & 0 & 0 & 0 & 0 & 0 \\ v/\epsilon c_t^2 & 0 & \rho & 0 & 0 & 0 & 0 \\ h + \frac{1}{2}(u^2 + v^2) & -1 & \rho u & \rho v & \rho C_p & \frac{5}{3}\rho & 0 \\ k/\epsilon c_t^2 & 0 & 0 & 0 & 0 & \rho & 0 \\ \epsilon/\epsilon c_t^2 & 0 & 0 & 0 & 0 & 0 & \rho \\ Y_i/\epsilon c_t^2 & 0 & 0 & 0 & 0 & 0 & \rho \end{pmatrix}$$

Note that we retain Q_v as the set of primary dependent variables.

Except for the preconditioning matrix and the choice of primary dependent variables, Eqn. (6.4) is identical to the standard governing equations in conservative form. The flux vectors E , F and H take their standard form. To solve flows on arbitrary grid configurations, Eqn. (6.4) is transformed to a generalized co-ordinate system. For the sake of brevity, these details are not presented.

6.3 Computational Results

6.3.1 Multi-Species Non-Reacting Shear Layers

To demonstrate the efficiency of the preconditioned system, we examine a multi-species, nonreacting and reacting shear-layer. The shear-layer consists of co-flowing streams of hydrogen and oxygen (Mach number of both streams is 0.1). Mild grid stretching is employed at the interface between the two streams in order to resolve the shear layer. The solutions for three different Reynolds number computations (200, 2000 and 20,000 based on the width of the oxygen stream) are shown in Fig. 6.1. For each case, different grid stretchings were employed yielding local maximum grid aspect ratios of 30, 60, and 100 respectively. The hydrogen profiles in Fig. 6.1 are obtained at a given axial location of the flowfield and show the extent of species diffusion for the three Reynolds numbers.

The convergence histories for the shear layer computations are shown in Fig. 6.2. For the enhanced algorithm (based on min-CFL, max-VNN and MOC II boundary conditions), the convergence for all three cases is seen to be very good reaching machine zero in about 1200 iterations. In particular, the different grid aspect ratios (shown in parenthesis) for the three cases have no effect on the convergence rate.

The other convergence curves shown in Fig. 6.2 are for the $Re = 200$ case with one or more aspects of the enhanced algorithm turned off. Curve II, for instance, is for the original algorithm (based on max-CFL, max-VNN). The corresponding convergence is about three times slower even though the grid aspect ratios for this case are not very high (the maximum grid aspect ratio is 30). Thus, even for this simple case, the savings in CPU time can be quite substantial.

The remaining convergence plots in Fig. 6.2 are obtained with the min-CFL time step definition but without viscous preconditioning (curve III) and without any preconditioning whatsoever (curve IV). These results once again demonstrate the importance of doing ‘everything’ right in order to obtain convergence that is independent of grid aspect ratio.

6.3.2 Reacting Shear Layers

We next consider results with the chemical reactions turned on. As mentioned previously, we employ a 9 species/18 reactions kinetics package for the chemistry [28]. Figure 6.3 shows the convergence for the $Re = 200$ case. The corresponding cold flow case was computed first and the reactions were then switched on. However, when the reactions were turned on to their full level, it proved impossible to obtain convergent solutions. Therefore, in the initial stages of the reacting computation, all the reaction rates (forward and backward) of the system were lowered and then gradually increased to the full value. This corresponds to the initial jagged region of the convergence plot. After the reactions are fully turned on, very good convergence is obtained. In fact, the convergence rate of the full reacting computation is very similar to the non-reacting case. Figure 6.4 shows the temperature and water mass fractions resulting from this computation. The maximum flame temperature is about 3400 K, which is characteristic of H_2/O_2 diffusion flames. These results also show that the chemical kinetics of the H_2/O_2 reacting system is extremely fast and there is virtually no observable flame stand-off distance.

The results of a similar computation for a $Re = 2000$ is shown in Figs. 6.5 and 6.6. The case was once again initiated with a cold flow computation and the reactions were turned on gradually. For this higher Reynolds number case, it proved to be somewhat more difficult to initialize the computation. However, once the reactions are turned on completely, the code converges once again at approximately the same rate as the cold flow computations. In these calculations, the chemical source term is treated fully implicitly and the above results show that this renders the system unconditionally stable once the flowfield has been initialized.

In the initial stages of the reacting computations, however, fast chemical kinetic rates tend to cause stability problems. These instabilities are related to non-linear effects that predominate during these initial stages. The difficulties may be overcome effectively by controlling the source term time scales at the initial stages of the computation [29]. This allows the solution to develop more gradually, thus allowing the flowfield to adapt to it in a stable manner. This may be accomplished in several ways—by controlling the reaction rates during the initial stages (as was done here) or by controlling the time-step sizes of the species transport equations independent of the fluid-dynamic equations. The latter procedure may be effected through preconditioning-based control of the system and is described in Ref. [29].

6.3.3 Turbulent Shear Layers

The next example that we consider is a multi-species, turbulent shear layer [30]. The $k - \epsilon$ model is used in addition to the fluid-dynamic and species transport equations as described earlier. Figure 6.7 shows the grid geometry (101 X 81) and axial velocity contours, while Fig. 6.8 shows the convergence rate for the computation. Again, in the initial stages of

the computation, the time-step size had to be lowered for stability. After about five hundred iterations, the time-steps were increased to their normal optimum values and very good convergence is obtained. Figures 6.9 and 6.10 show comparisons of the predicted velocity profiles and turbulent kinetic energy profiles with experimental data [30]. Reasonably good agreement is observed. The discrepancies in the turbulent kinetic energy near the edges of the shear layer are probably due to intermittency associated with large-scale unsteadiness.

6.3.4 Turbulent Flow Over a Backward-facing Step

The final example that we consider is the turbulent flow over a two-dimensional backward step, that was experimentally studied by Driver and Seegmiller [31]. Figure 6.11 shows the computational grid (121 X 121) and axial velocity contours. The predicted recirculation length normalized by the step-height was 5.5, while the experimental value is 6.3. Interestingly, this case proved to be unstable when standard central-difference with fourth-order artificial dissipation was used. Examination of the residuals suggest that the source of the instability was the unsteady vortex-shedding processes associated with the shear layer at the edge of the backward step. In order to stabilize the computation, additional second-order dissipation is necessary and was introduced using a Jameson-Turkel switch [32]. The switched dissipation was based on the gradient of the velocity so that the vortex-shedding was suppressed at the shear-layer, while being negligibly small at other regions of the flowfield. Figure 6.12 shows the associated convergence rate. The residuals are observed to go down six to seven orders in magnitude in about 2000 iterations.

Comparison of the computed velocity and turbulent kinetic energy profiles with experimental data are shown in Fig. 6.13 and 6.14 respectively. In addition to the central-differenced scheme described above, results are also shown for several upwind-based treatments (upwind schemes are considered in detail in the following section). The inherent dissipation of the upwind schemes appear to be sufficient to stabilize the computations without any additional dissipation for this case. The velocity profiles (in Fig. 6.13) agrees very well with the experimental data, while the turbulent kinetic energy profiles (in Fig. 6.14) show modest agreement. Again, it may be observed that the turbulence levels are under-predicted at the edge of the shear layer and is probably due to flow intermittency associated with large scale eddies.

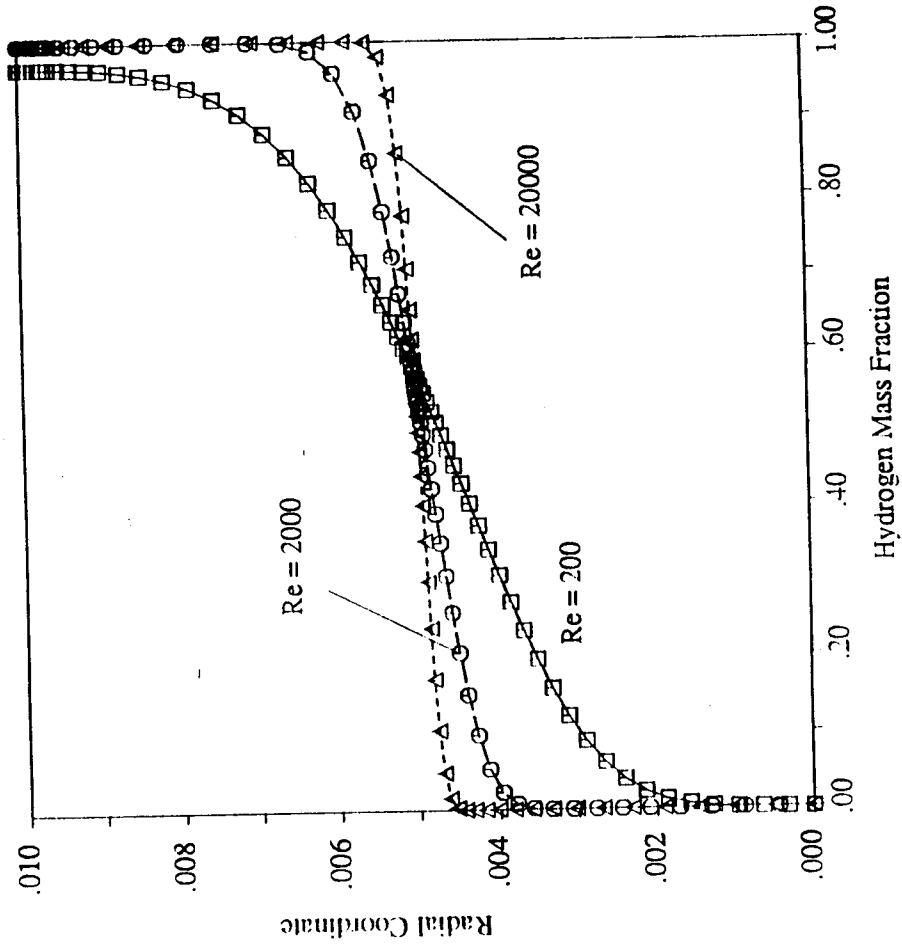


Fig. 6.1. Shear layer hydrogen mass fraction profiles.

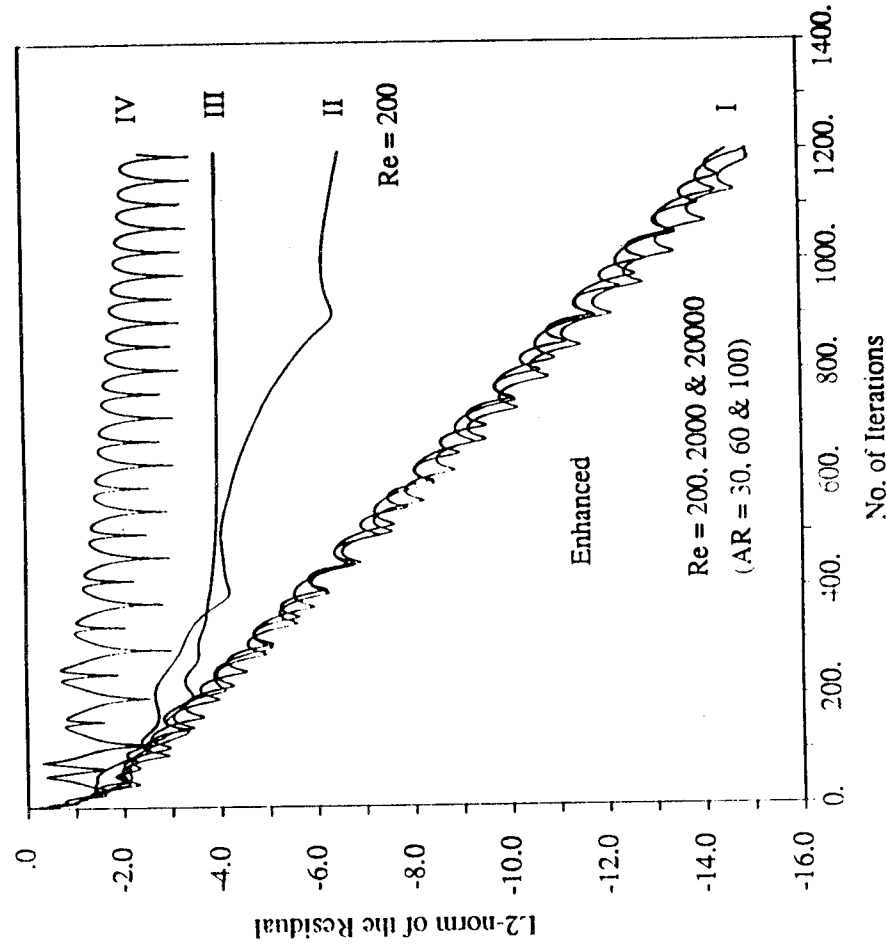


Fig. 6.2. Convergence for multi-species shear layer for standard and enhanced algorithms.
 I - min-CFL/max-VNN and MOC II boundary conditions with viscous preconditioning; II
 - max-CFL/max-VNN, MOC II, viscous preconditioning; III - min-CFL/max-VNN, MOC
 II, with inviscid preconditioning but no viscous preconditioning; IV - no preconditioning.

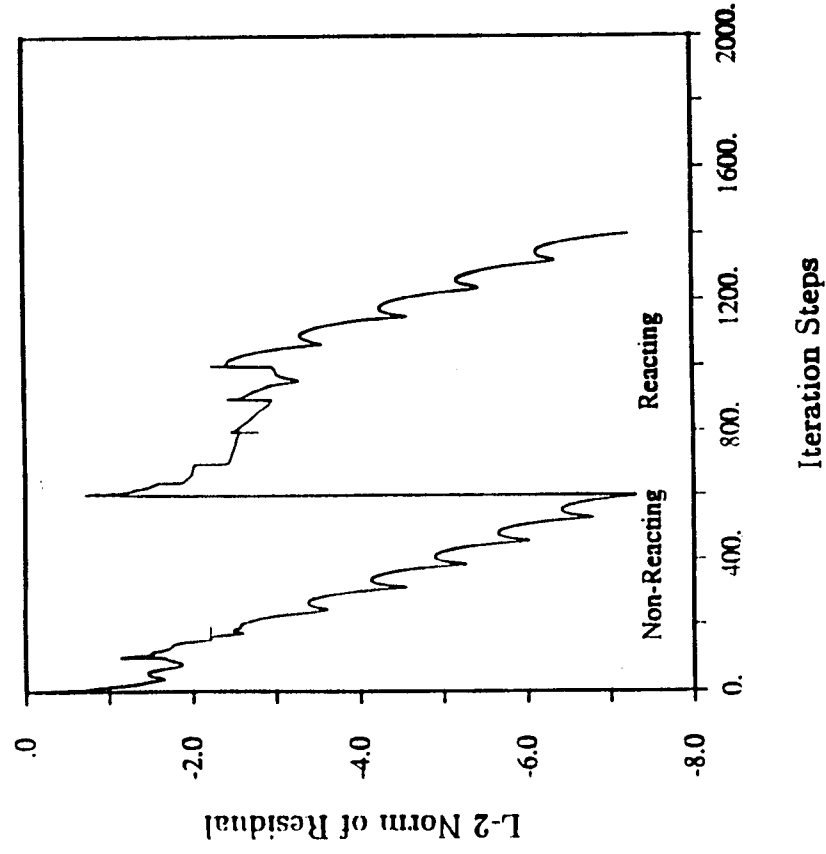
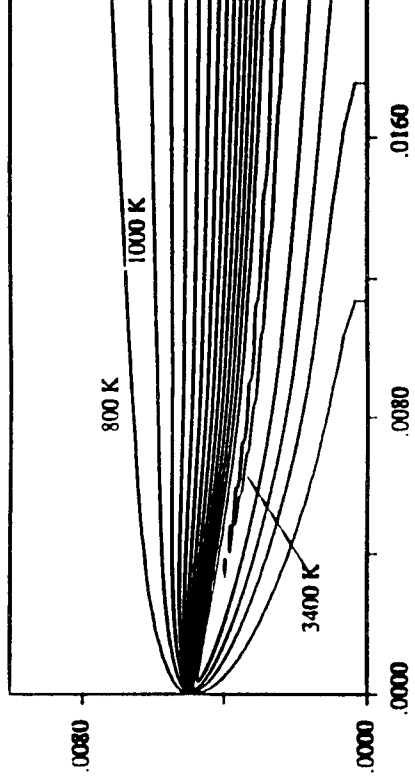


Fig. 6.3. Convergence of preconditioned code for reacting shear layers. ($Re=200$).

Temperature



Water Mass Fractions

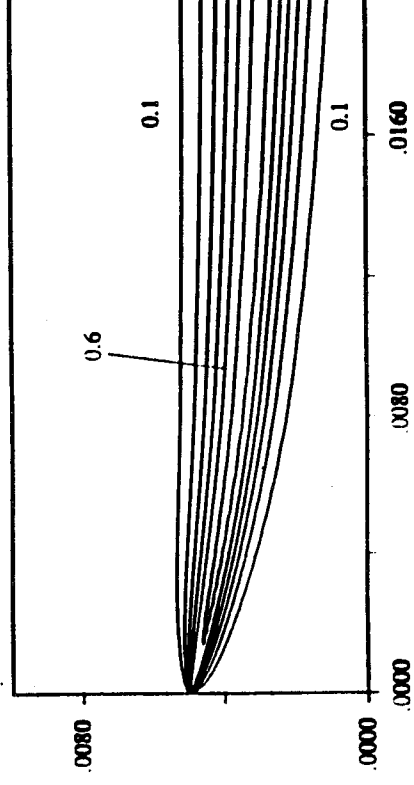


Fig. 6.4. Converged temperature and water mass fractions for reacting shear layer.
($Re=200$).

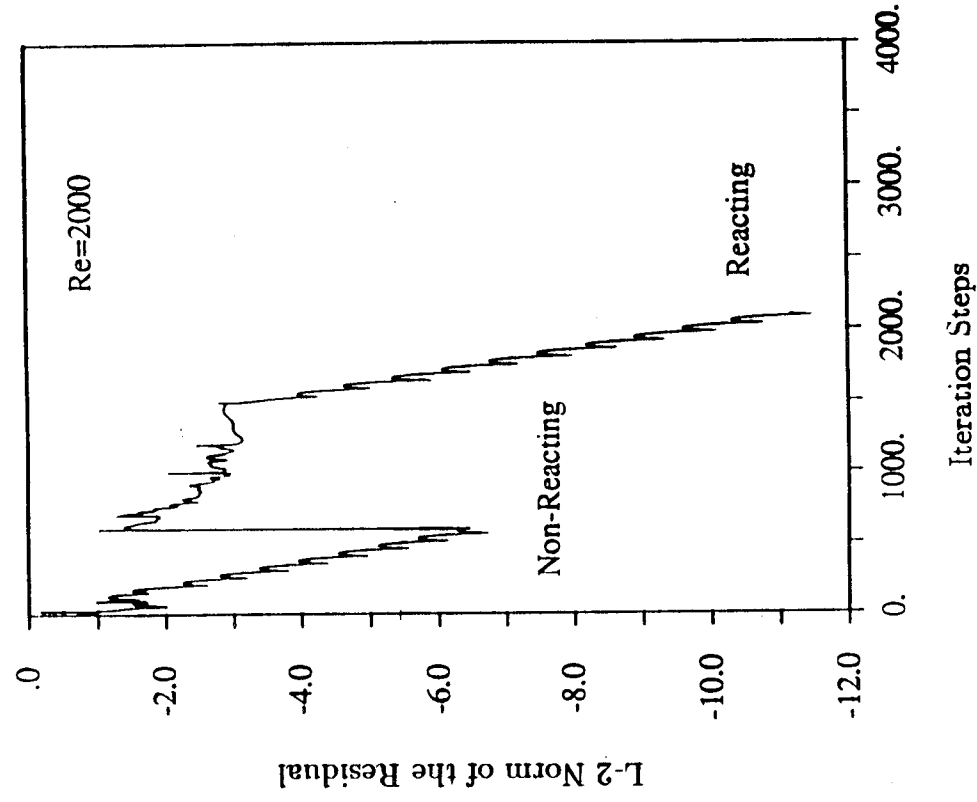


Fig. 6.5. Convergence of preconditioned code for reacting shear layers. ($Re=2000$).

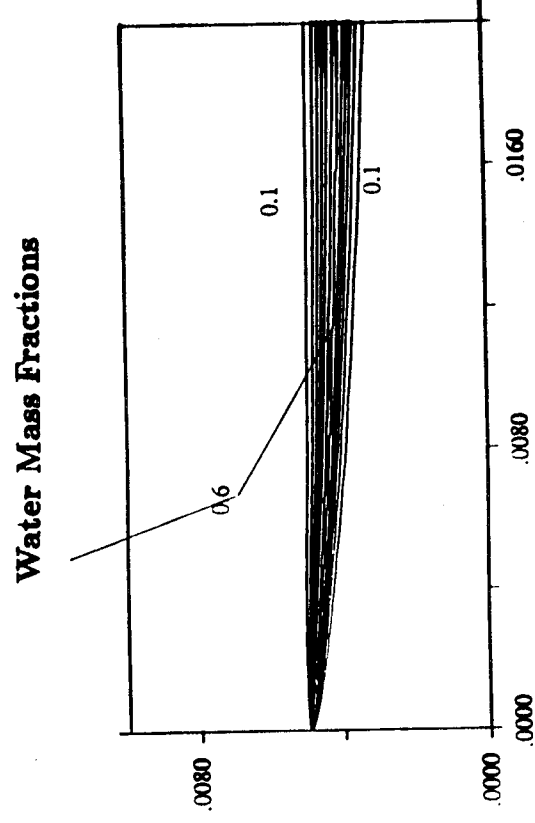
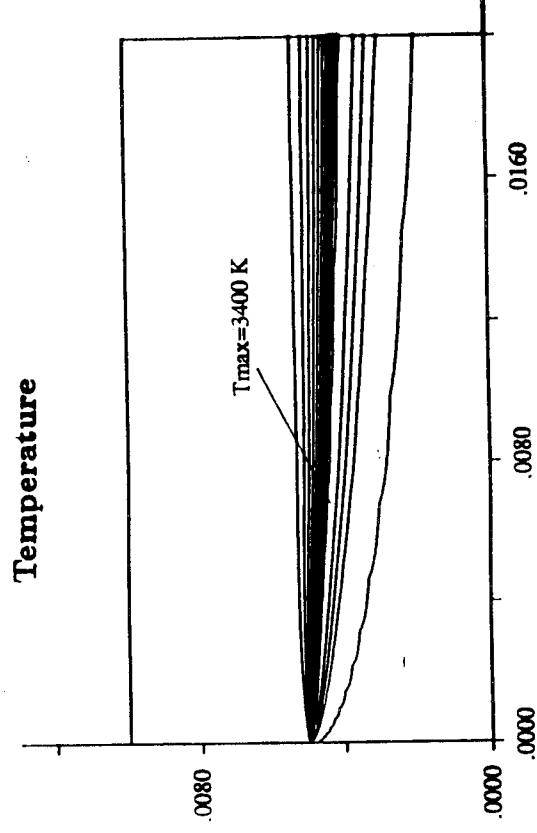


Fig. 6.6. Converged temperature and water mass fractions for reacting shear layer.
($Re=2000$).

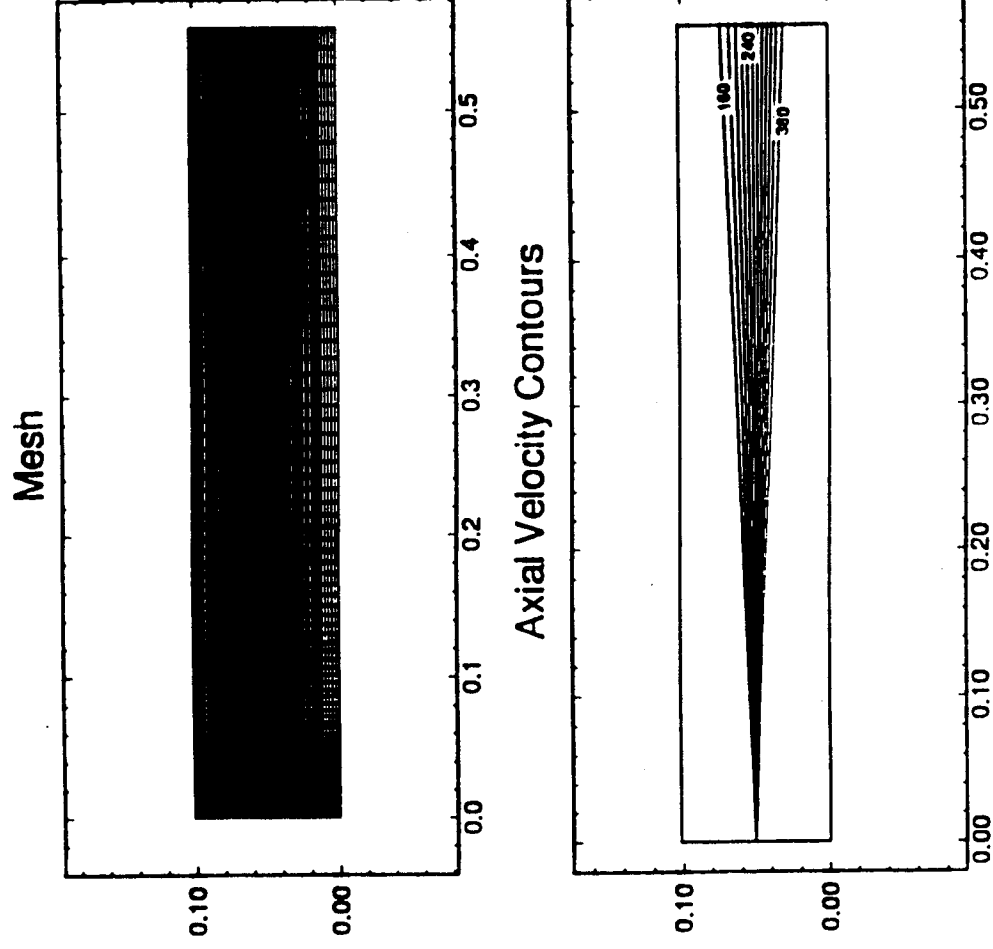


Fig. 6.7. Grid and axial velocity contours for a multi-species, turbulent shear layer.

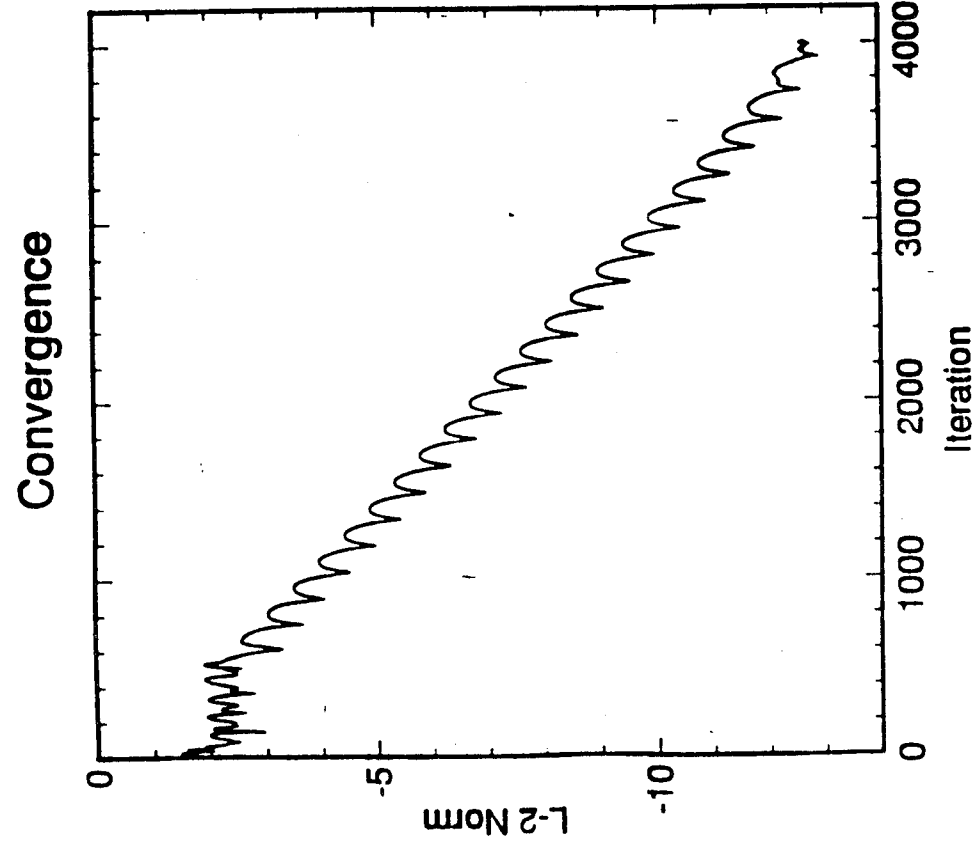


Fig. 6.8. Convergence of preconditioned code for multi-species, turbulent shear layer.

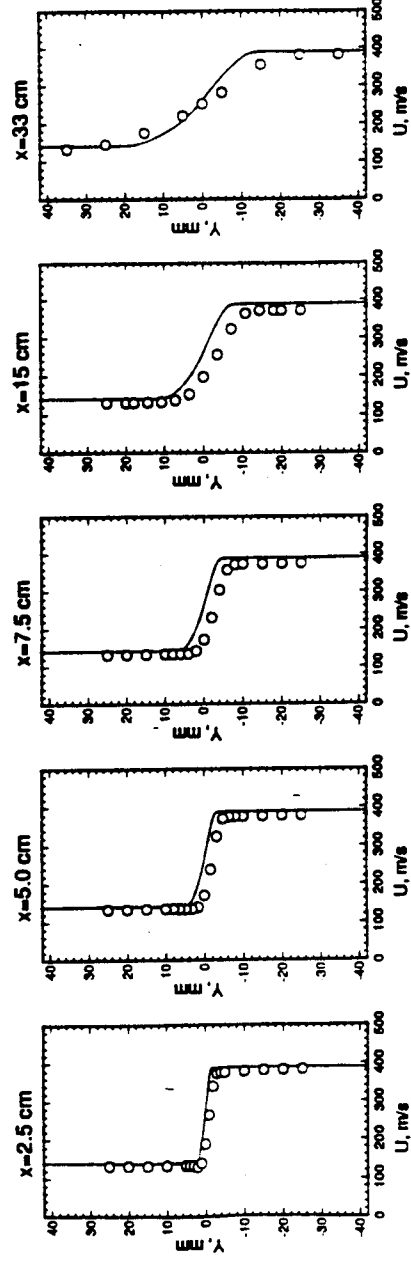


Fig. 6.9. Comparison of predicted velocity profiles with the experimentally determined values of Ref. [29] at various downstream locations for the multi-species, turbulent shear layer.

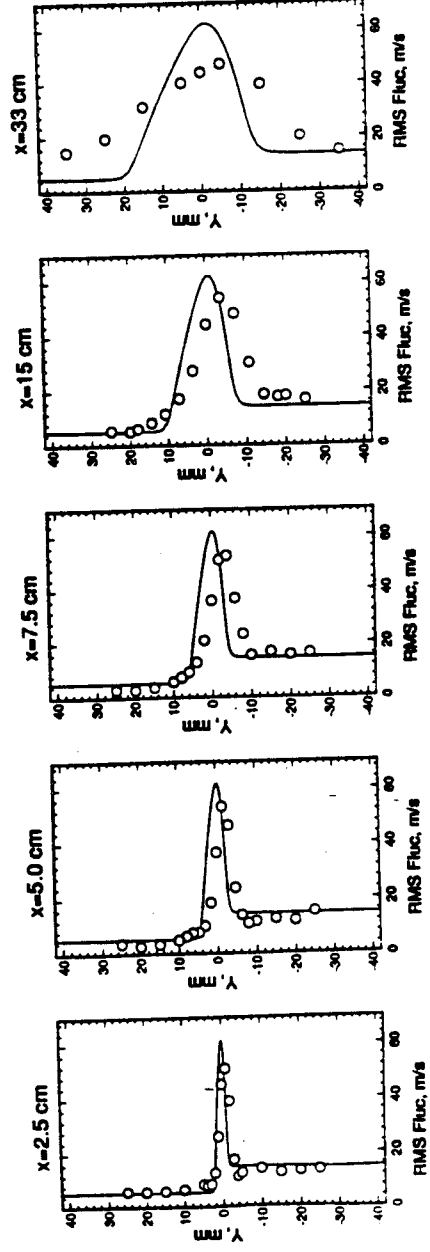


Fig. 6.10. Comparison of predicted turbulent kinetic energy profiles with the experimentally determined values of Ref. [29] at various downstream locations for the multi-species, turbulent shear layer.

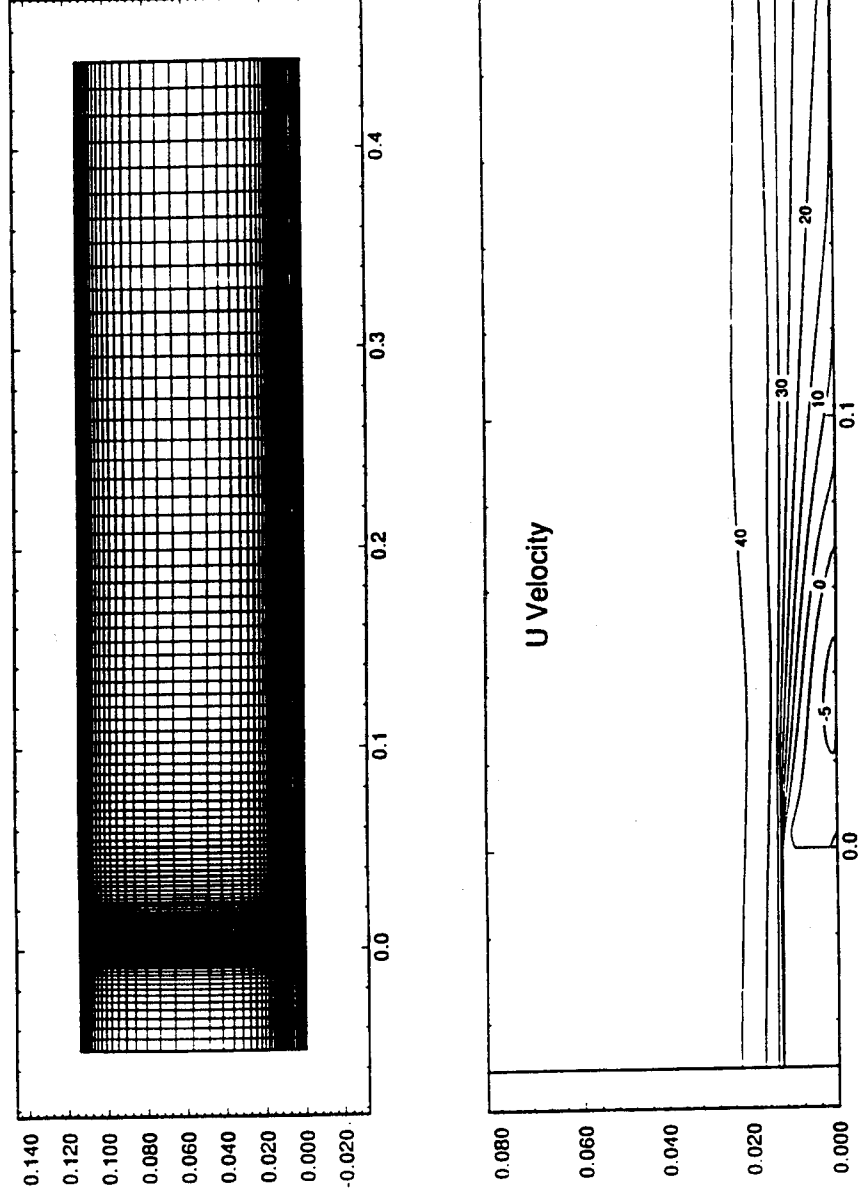


Fig. 6.11. Grid and axial velocity contours for turbulent flow over a backward-facing step.

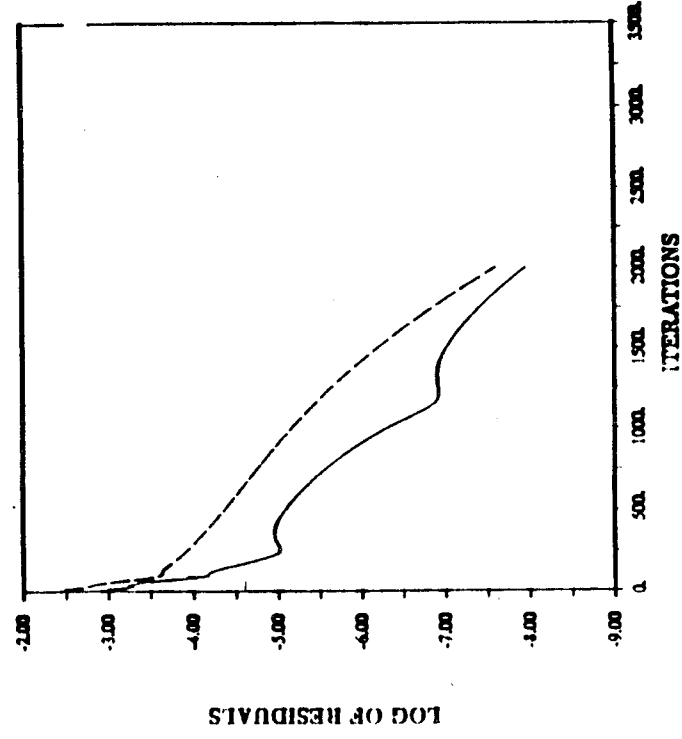


Fig. 6.12. Convergence histories for the turbulent flow over a backward-facing step.

Backstep Flow - Velocity Profiles

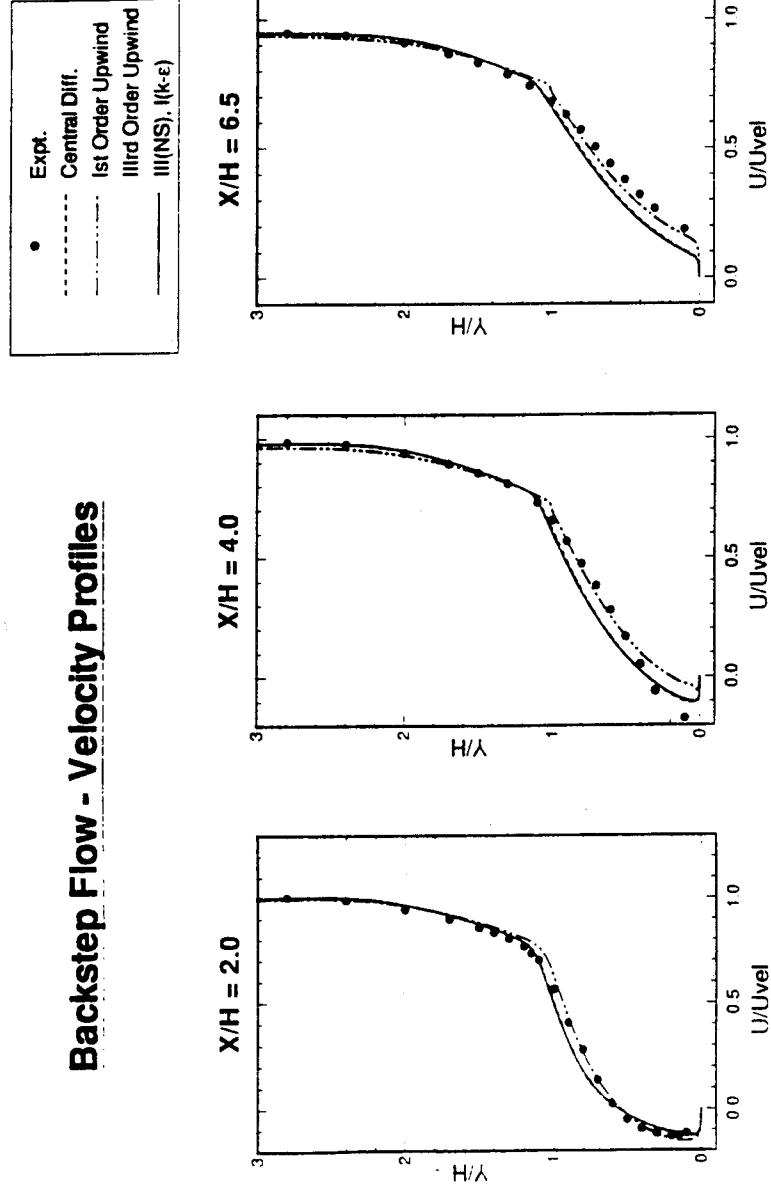


Fig. 6.13. Comparison of predicted velocity profiles with the experimentally determined values of Ref. [30] at various downstream locations.

Backstep Flow - Turbulent KE Profiles

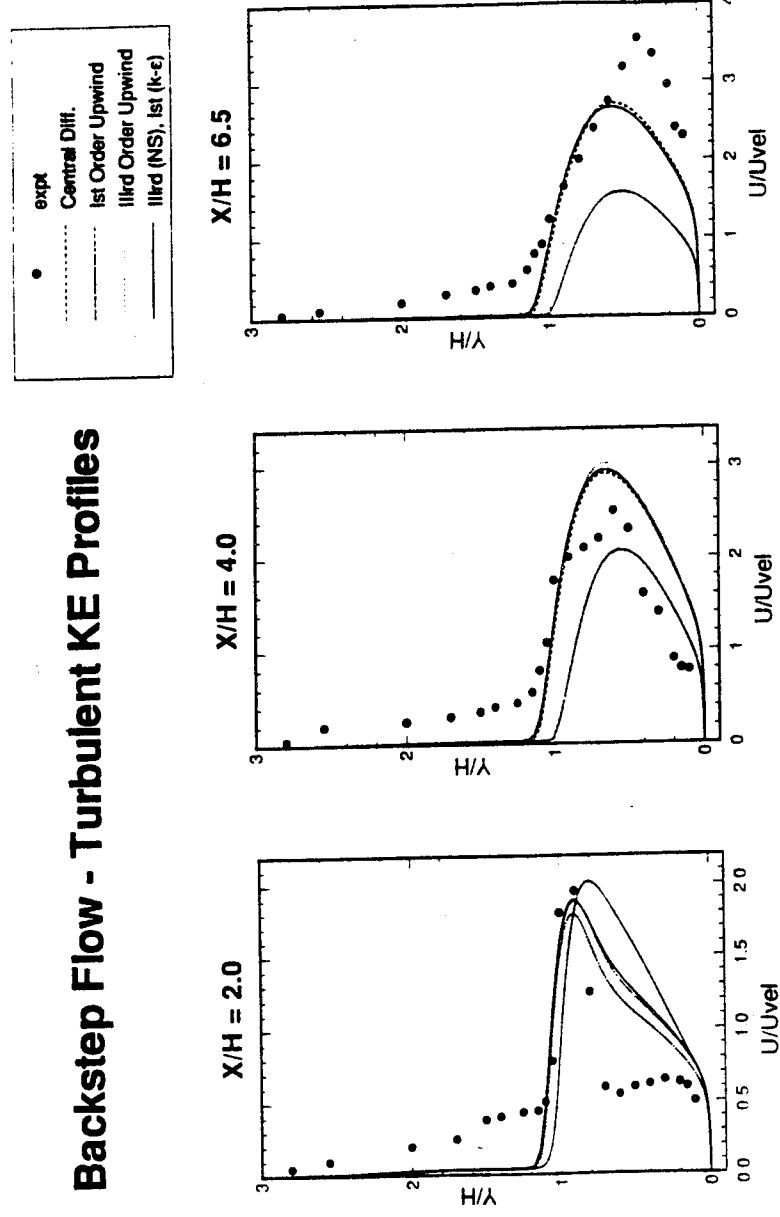


Fig. 6.14. Comparison of predicted turbulent kinetic energy profiles with the experimentally determined values of Ref. [30] at various downstream locations.

APPLICATION OF PRECONDITIONING TO UPWIND SCHEMES

Upwind schemes, by virtue of their inherent diagonal dominance, allow a wide variety of approximate methods to be used for inverting the implicit LHS operator. The ADI approximate factorization method again represents a viable choice [26, 33], as do the point Gauss-Seidel (or LU) [33, 34, 35] and line Gauss-Seidel (LGS) [33, 36, 37, 38, 39, 40] methods. All of these techniques work well for simple transonic flow Euler problems on square grids. Researchers typically choose one of these schemes based either on familiarity, vectorizability or parallelizability considerations. However, the performance of these schemes in the presence of high grid aspect ratios has not been systematically evaluated in a comparative sense.

This section is organized as follows. We start by presenting the general form of the Navier-Stokes equations that we have employed in all of our studies and apply the upwind flux-difference split algorithm to them. We go on to summarize the various factorization/relaxation procedures that can be used for approximately inverting the implicit operator. This is followed by an evaluation of the stability domains of the schemes and, finally, by numerical computations of simple flowfields to support the stability results. Specific attention is addressed to first the Euler and then the Navier-Stokes equations. In the last section, we summarize the conclusions drawn from the stability and convergence studies.

7.1 Formulation of Upwind Schemes

7.1.1 Flux-Difference Splitting Procedure

The convective fluxes are upwinded by standard flux-difference splitting methods. The numerical representation of the convective fluxes at the cell faces is defined by:

$$\hat{E}_{i+1/2} = \frac{1}{2} [E_{i+1} + E_i] - \hat{D}_{i+1/2} \quad (7.1)$$

with similar definitions for $\hat{E}_{i-1/2}$ and $\hat{F}_{j\pm 1/2}$. The artificial diffusivity, $\hat{D}_{i+1/2}$, is given by:

$$\begin{aligned} \hat{D}_{i+1/2} = & k_1 [\Delta E_{i+1/2}^+ - \Delta E_{i+1/2}^-] - k_2 [\Delta E_{i-1/2}^+ - \Delta E_{i+3/2}^-] \\ & - k_3 [\Delta E_{i+3/2}^+ - \Delta E_{i-1/2}^-] \end{aligned} \quad (7.2)$$

where $(k_1, k_2, k_3) = (1/2, 0, 0)$ gives a first-order upwind flux, $(k_1, k_2, k_3) = (1/6, 1/6, 0)$ provides a third-order upwind-biased flux, and $(k_1, k_2, k_3) = (1/4, 1/8, 1/8)$ gives a second-order central plus a third-order matrix dissipation. The standard central-difference representation

is obtained by setting $(k_1, k_2, k_3) = (0, 0, 0)$ and adding fourth-order scalar dissipation. In the above expressions,

$$\Delta E_{i\pm 1/2}^\pm = \left(\Gamma M_x \Lambda_x^\pm M_x^{-1} \right)_{i\pm 1/2} \Delta Q_{v_{i\pm 1/2}}$$

and

$$\Gamma^{-1} A = M_x \Lambda_x M_x^{-1}$$

where $\Delta Q_{v_{i+1/2}} = Q_{v_{i+1}} - Q_{v_i}$, M_x and M_x^{-1} are the modal matrices which diagonalize $\Gamma^{-1} A$, and $A = \partial E / \partial Q_v$. Similar definitions can be given for $\Delta F_{j\pm 1/2}^\pm$. To simplify the presentation we also define the matrices:

$$A_i^\pm = \frac{1}{2} \left(A_i \pm \Gamma |\Gamma^{-1} A|_{i\pm 1/2} \right), \quad B_j^\pm = \frac{1}{2} \left(B_j \pm \Gamma |\Gamma^{-1} B|_{j\pm 1/2} \right)$$

Combining these definitions with the diffusion terms, and using Euler implicit differencing in time we obtain the fully-implicit delta-form:

$$\left[\Gamma + \frac{\Delta t}{\Delta x} \left(\nabla_x A^+ + \Delta_x A^- \right) + \frac{\partial T}{\partial y} \left(\nabla_y B^+ + \Delta_y B^- \right) - \Delta t \left(\frac{\partial}{\partial x} R_{xx} \frac{\partial}{\partial x} + \frac{\partial}{\partial y} R_{yy} \frac{\partial}{\partial y} \right) \right] \Delta Q_v = -\Delta t \mathcal{R}^n \quad (7.3)$$

where Δ_x and ∇_x represent first-order forward, and first-order backward differencing in x , with similar definitions for the Δ_y and ∇_y operators. The quantity $\mathcal{R}^n$ represents the residual of the steady-state equations,

$$\mathcal{R}^n = \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} - \frac{\partial}{\partial x} R_{xx} \frac{\partial Q_v}{\partial x} - \frac{\partial}{\partial x} R_{xy} \frac{\partial Q_v}{\partial y} - \frac{\partial}{\partial y} R_{yx} \frac{\partial Q_v}{\partial x} - \frac{\partial}{\partial y} R_{yy} \frac{\partial Q_v}{\partial y}$$

The LHS operator is treated using first-order upwind differencing for the convective terms, while the right-hand-side (RHS) residual is typically treated with second-order central or third-order upwind-biased representations. The diffusive terms are modeled with standard central differences.

7.1.2 Approximate-Inversion Methods

Efficient solution of Eqn. (7.3) requires that it be solved by some approximate means. The diagonally dominant form of the LHS allows numerous such methods to be used. Here we consider three common algorithms (along with minor modifications as noted below): the two-sweep point Gauss-Seidel method (LU factorization), the two- or four-sweep line Gauss-Seidel (LGS-2 and LGS-4) method, and the upwind implementation of the alternating-direction implicit (ADI) method. To simplify the presentation and reduce the amount of

algebra, these schemes will be presented only for the Euler equations. Extension to the Navier-Stokes equations is straightforward.

The two-sweep point Gauss-Seidel algorithm gives rise to a lower triangular matrix on the first sweep, and an upper triangular matrix on the second, thus giving an LU approximately factored system.

$$\left[S - \frac{\Delta t}{\Delta x} \mathcal{A}_{i-1}^+ - \frac{\partial T}{\partial y} \mathcal{B}_{j-1}^+ \right] \Delta Q_v^* = -\Delta t \mathcal{R}^n \quad (7.4a)$$

$$\left[S + \frac{\Delta t}{\Delta x} \mathcal{A}_{i+1}^- + \frac{\partial T}{\partial y} \mathcal{B}_{j+1}^- \right] \Delta Q_v = -\Delta t \mathcal{R}^n + \frac{\Delta t}{\Delta x} \mathcal{A}^+(\Delta Q_v^*)_{i-1} + \frac{\partial T}{\partial y} \mathcal{B}^+(\Delta Q_v^*)_{j-1} \quad (7.4b)$$

where

$$S = \Gamma + \frac{\Delta t}{\Delta x} (\mathcal{A}_i^+ - \mathcal{A}_i^-) + \frac{\partial T}{\partial y} (\mathcal{B}_j^+ - \mathcal{B}_j^-)$$

This grouping of terms on the diagonal makes the scheme diagonally dominant. The LU scheme requires only block diagonal inversions. Occasionally, researchers employ multiple sweeps through the domain at each time step as discussed later.

The line Gauss-Seidel (LGS) algorithm requires block tri-diagonal inversions along a line across the domain. The LGS algorithm can be applied in a two- or four-sweep manner. The four-sweep LGS algorithm, for instance, is given as:

Forward x -sweep

$$\left[S - \frac{\partial T}{\partial y} (\mathcal{B}_{j-1}^+ - \mathcal{B}_{j+1}^-) - \frac{\Delta t}{\Delta x} \mathcal{A}_{i-1}^+ \right] \Delta Q_v^* = -\Delta t \mathcal{R}^n \quad (7.5a)$$

Backward x -sweep

$$\left[S - \frac{\partial T}{\partial y} (\mathcal{B}_{j-1}^+ - \mathcal{B}_{j+1}^-) + \frac{\Delta t}{\Delta x} \mathcal{A}_{i+1}^- \right] \Delta Q_v^{**} = -\Delta t \mathcal{R}^n + \frac{\Delta t}{\Delta x} \mathcal{A}_{i-1}^+ (\Delta Q_v^*)_{i-1} \quad (7.5b)$$

Forward y -sweep

$$\left[S - \frac{\Delta t}{\Delta x} (\mathcal{A}_{i-1}^+ - \mathcal{A}_{i+1}^-) - \frac{\partial T}{\partial y} \mathcal{B}_{j-1}^+ \right] \Delta Q_v^{***} = -\Delta t \mathcal{R}^n - \frac{\partial T}{\partial y} \mathcal{B}_{j+1}^- (\Delta Q_v^{**})_{j+1} \quad (7.5c)$$

Backward y -sweep

$$\left[S - \frac{\Delta t}{\Delta x} (\mathcal{A}_{i-1}^+ - \mathcal{A}_{i+1}^-) + \frac{\partial T}{\partial y} \mathcal{B}_{j+1}^- \right] \Delta Q_v = -\Delta t \mathcal{R}^n + \frac{\partial T}{\partial y} \mathcal{B}_{j-1}^+ (\Delta Q_v^{***})_{j-1} \quad (7.5d)$$

where S is as previously defined.

Variations of the LGS algorithm include a single forward/backward sweep for each coordinate direction or multi-sweep schemes where several sweeps are performed at each time-level. Other variations of the four-sweep scheme may also be obtained by re-ordering the sweeps. A two-sweep LGS scheme is obtained by dropping Eqs. (7.5c) and (7.5d).

The upwind ADI algorithm takes the form of an x -sweep followed by a y -sweep,

$$\left[\Gamma + \frac{\Delta t}{\Delta x} (\nabla_x \mathcal{A}^+ + \Delta_x \mathcal{A}^-) \right] \Delta Q_v^* = -\Delta t \mathcal{R}^n \quad (7.6a)$$

$$\left[\Gamma + \frac{\partial T}{\partial y} (\nabla_y \mathcal{B}^+ + \Delta_y \mathcal{B}^-) \right] \Delta Q_v = \Gamma \Delta Q_v^* \quad (7.6b)$$

If all diagonal terms are grouped together before factoring, the resulting scheme would be a “diagonally-dominant” ADI (or DDADI) scheme. Analysis of the DDADI scheme reveals poor overall stability and convergence as opposed to the above (non-diagonally dominant) ADI scheme. In contrast, the non-diagonally dominant form of the LU scheme (not shown here) yields more restrictive stability than the diagonally-dominant LU scheme presented above.

Note that the above relaxation schemes can also be written as approximate factorization schemes. Here we note the approximate factorization errors for the LU, two-sweep LGS, and upwind ADI schemes:

LU:

$$- \left(\frac{\Delta t}{\Delta x} \mathcal{A}_{i-1}^+ + \frac{\partial T}{\partial y} \mathcal{B}_{j-1}^+ \right) S^{-1} \left(\frac{\Delta t}{\Delta x} \mathcal{A}_{i+1}^- + \frac{\partial T}{\partial y} \mathcal{B}_{j+1}^- \right) \quad (7.7)$$

Two-Sweep LGS:

$$\frac{\Delta t^2}{\Delta x^2} \mathcal{A}_{i-1}^+ \left[S - \frac{\partial T}{\partial y} (\mathcal{B}_{j-1}^+ - \mathcal{B}_{j+1}^-) \right]^{-1} \mathcal{A}_{i+1}^- \quad (7.8)$$

ADI:

$$\frac{\Delta t^2}{\Delta x \Delta y} (\nabla_x \mathcal{A}^+ + \Delta_x \mathcal{A}^-) \Gamma^{-1} (\nabla_y \mathcal{B}^+ + \Delta_y \mathcal{B}^-) \quad (7.9)$$

Because the product $\mathcal{A}_i^+ \mathcal{A}_i^-$ is identically zero, we see that the LGS method produces an approximate factorization error (Eq. 7.8) that would vanish completely (implying the approximate factorization is an exact factorization) if the matrix $\mathcal{A}_{i-1}^+$ and the matrix in brackets commuted, and if $\mathcal{A}_{i-1} = \mathcal{A}_{i+1}$ (*i.e.* no spatial variation in $\mathcal{A}$). In practice these matrices are not commutative, and the matrix $\mathcal{A}$ varies in space, but this approximate factorization error does tend to be small as later results suggest. Similarly, some (but not all) of the error terms in the LU system (Eq. 7.9) come close to cancelling. Note, however, that even for the direct inversion case we only approach a Newton iteration when the LHS and RHS employ consistent spatial differencing.

7.2 Results

Stability analyses of numerical schemes have two distinctly different uses. First of all, they predict the stability or instability of a given numerical algorithm, but perhaps more importantly, they also predict the rate of decay of errors, and therefore the convergence rate of the scheme. Stability analyses of upwind Euler and Navier-Stokes solution methods are particularly important because there are so many competing methods to choose from. The key issue is to be able to anticipate the relative performance of the various schemes on a variety of grid sizes and for a wide range of physical conditions before choosing a final algorithm. Naturally, as with any analytical model, it is imperative to test the theoretical predictions against experiment to verify their authenticity. This experimental testing is reported following the stability results.

One surprising conclusion that is reached after assessing the stability characteristics of upwind algorithms for the Euler equations is that the LU and LGS methods are conditionally stable and must observe a maximum time step restriction. Extrapolation from scalar applications has often led to the incorrect assumption that these methods are unconditionally stable. However, we note that particularly the LGS method provides very efficient convergence despite its conditional stability. Consequently, appropriate choice of the time-step size is particularly important for these classes of schemes.

A second surprising finding is that the ADI method behaves almost exactly the same when upwind differencing is used as it does when central differencing is used. The upwind ADI system remains unconditionally stable, has the same optimum CFL, and converges at essentially the same rate for upwind systems as for central difference systems.

Von Neumann stability analyses of the schemes presented in Eqs. (7.3)-(7.6) are first carried out for the Euler equations and then for the Navier-Stokes equations. Convergence analysis of the Euler equations is important for both Euler and high Reynolds number Navier-Stokes solutions since at high cell Reynolds numbers, the effects of diffusion can be ignored and the stability and convergence of the full Navier-Stokes system is identical to that for the Euler equations. Nevertheless, an assessment of the stability of the Navier-Stokes equations is equally important because even for high Reynolds number problems, the effects of diffusion begin to dominate near walls (as they do throughout the flowfield for low Reynolds number applications) and their local effects must be considered. Following the stability analyses, convergence results from inviscid and viscous example problems are presented to support the stability findings.

7.2.1 Euler Stability Analysis

The proper selection of the local time-step size (Δt) is crucial for the efficient application of any CFD algorithm. We have shown earlier that the optimum local time step for the 2-D central-differenced ADI algorithm is given by the “min-CFL” definition (Eqn. 5.2). So long as the particle and acoustic speeds are of comparable magnitudes (i.e., at transonic speeds), this definition provides uniform convergence for all grid aspect ratios. Note that this choice implies the minimum CFL at every grid point is set to the “optimum” value, while the maximum CFL (which differs from the minimum value by the grid aspect

ratio ($AR=\Delta x/\Delta y$) may become very large. For low Mach numbers, this choice optimizes acoustic CFL's while the particle CFL is extremely small, leading once again to poor convergence rates. In such cases, preconditioning becomes necessary. The definition of the preconditioning matrix was discussed earlier and is given in Eqn. (2.3). As we will see, these considerations of optimal time-step selection and preconditioning remain crucial for ensuring good convergence behaviour of upwind methods as well.

7.2.1.1 Direct Inversion

Before we consider the stability of the various approximate methods, we start with the direct inversion procedure in Eq. (7.3). This (presumably) provides faster iterative convergence than can be attained by any of the approximate methods. The stability results presented here as well as those in the remainder of Section VII are for the first-order upwind LHS, third-order RHS (referred to as I/III). In general, the results for the I/II scheme (first-order LHS, second-order RHS) are very similar to the I/III results, while I/I results tend to be much more favorable. These differences are discussed where appropriate.

Figure 7.1 shows the maximum eigenvalues of the amplification matrix in wave-number space for $M=0.7$, flow angle of 45 degrees and $AR=1$. Because these stability results are symmetric about the leading diagonal, results are shown for two different CFL's (10 and 1000), one on each half. For both CFL's, good damping is evident over the entire wave-number domain with the higher CFL suggesting slightly faster convergence. At even higher CFL's, the stability plot becomes invariant, demonstrating the scheme's unconditional stability, but indicating that no further gains in convergence rate can be realized by increasing the time step. This 'freezing' of the stability diagram is a consequence of the inconsistent differencing of the LHS and RHS. In contrast, for consistent (*e.g.* I/I) differencing, increasing the CFL causes the amplification factors to progressively approach zero over the entire wave-number domain, indicating correspondingly faster convergence rates as Δt is increased.

Figure 7.2 shows similar stability eigenvalues but for $M = 0.001$. Results are shown for $CFL=1000$ both without (upper half) and with (lower half) preconditioning. Without preconditioning at this low Mach number, the amplification factor is seen to be somewhat stiff (close to unity) for low wave-numbers, indicating slow convergence. Increasing the CFL further does not alleviate this stiffness problem because of the freezing of the stability map mentioned earlier. With preconditioning, on the other hand, excellent damping is observed at all wave-numbers, indicating rapid convergence behavior. This result demonstrates that low-Mach number preconditioning is effective even for the direct inversion scheme, a result that arises because of the inconsistent differencing of the LHS and RHS.

The stability map for the same conditions ($M=0.7$, $CFL=1000$) but for a larger aspect ratio ($AR=100$) is shown in Fig. 7.3. Here the min-CFL definition (Eq. 5.2) is used, and the directional CFL's are $CFL_x=1000$ and $CFL_y=10^5$. Because of the dissimilar CFL's in the two directions, the stability plot is no longer symmetric about any line, and the full plot is shown in Fig. 7.3. At this condition, the direct scheme shows good damping for all modes, including the purely axial modes along the x -axis, which are notoriously responsible for high aspect ratio convergence difficulties [41]. Consequently, the min-CFL time stepping is effective for the direct inversion, implicit upwind system.

7.2.1.2 Point Gauss-Seidel (LU) Scheme

All approximate methods for solving the LHS of Eq. (7.3) result in stability characteristics that are considerably different from those of the direct inversion. One primary difference that the LU scheme exhibits is the existence of an upper limit on the time step. Although this conditional stability limit depends on the flow Mach number and the flow angle, its effect is well-characterized by the min-CFL definition. Figure 7.4 shows the demarcation between stable and unstable time steps (min-CFL) for a wide range of cell aspect ratios. As can be seen, the limiting min-CFL value is around 3 for $AR=1$ but decreases to about 0.1 as the cell aspect ratio increases. In general, these limiting CFL's become more restrictive at low Mach numbers (without preconditioning), but with low-speed preconditioning, the results in Fig. 7.4 are valid for all speeds. We note in passing that this conditional stability is also observed for the I/II LU system, while the consistent I/I LU system retains the unconditional stability of the direct inversion procedure. The difference between the LU and the direct method arises from the approximate factorization errors (Eq. 7.7).

An example of the unstable eigenvalues is given in Fig. 7.5 which shows the stability map for the I/III scheme for $M=0.7$, $AR=100$ and $\min-CFL=10$. At this condition, the scheme is observed to be unstable for purely axial modes. Choosing $CFL=0.1$ (not shown) stabilizes the scheme, but because of the low CFL value, the amplification factor becomes stiff for all axial wave-numbers, indicating poor convergence. Note that the stability restrictions associated with the LU scheme occur only when inconsistent differencing is used between the LHS and RHS.

The multiple point-Gauss-Seidel sweeps improve the stability domain over the two-sweep results discussed above. This is because the point Gauss-Seidel inner or sub iterations at a given time-step are unconditionally stable. With four Gauss-Seidel sweeps (four lower and four upper), the stability plot in Fig. 7.5 is stabilized but it is still somewhat stiff for low wave-number axial modes. This stiffness may be alleviated by increasing the number of Gauss-Seidel sweeps even further—for this case, about 20 sweeps appear to be sufficient. However, note that the CFL number for this case was 10. To ensure unconditional stability (i.e., infinite CFL) of the scheme at high aspect ratios a significantly larger number of inner iterations is necessary. Thus, it appears that the point Gauss-Seidel scheme even with multiple sweeps is not well suited to handling high aspect ratio grid cells.

7.2.1.3 Line Gauss-Seidel (LGS) Scheme

We consider here the stability results for the four-sweep LGS (LGS-4) scheme (Eq. 7.5). Figure 7.6 shows the stability map for $M=0.7$, $AR=1$ and I/III differencing. Again, taking advantage of the symmetry along the leading diagonal, we present the results for both $CFL=10$ and $CFL=100$. The close similarity of the $CFL=10$ results with those of the direct scheme (Fig. 7.1) suggests that the LGS-4 scheme approximates the inverse of the LHS operator very well and should provide very good convergence. However, at the higher CFL of 100, the scheme becomes unstable at low wave-numbers unlike the direct scheme. The demarcation between stable and unstable time steps for the LGS-4 scheme (based on the min-CFL definition) is again shown in Fig. 7.4 for a wide range of grid aspect ratios. Again

the results show that the min-CFL definition correlates the stable/unstable boundary very well. The LGS algorithm is seen to be conditionally stable with a limiting CFL of about 20 for all grid aspect ratios and all subsonic Mach numbers with or without preconditioning.

Surprisingly, the stability of the I/I scheme also reveals a similar stability constraint. Thus, the source of the instability does not arise from the inconsistent differencing as was the case for the point Gauss-Seidel scheme. Analysis of the stability of the line Gauss-Seidel inner iterations likewise reveals a conditional stability criterion. These results appear counter to conventional wisdom since it is generally accepted that the procedure is unconditionally stable as long as the matrix operator is diagonally dominant. Careful examination, however, reveals that the line Gauss-Seidel procedure is unconditionally stable if it is composed of forward sweeps only or of reverse sweeps only. The combination of forward and reverse sweeps introduces the conditional stability. Maintaining a single sweep direction, however, is not practical since, although it is unconditionally stable, the stability map shows that it is stiff and converges poorly, a result that is due to the part-explicit treatment of the LHS operator.

Figure 7.7 shows the stability plot for the same $M=0.7$ case, but for an aspect ratio (AR) of 100. These results are for a min-CFL=10 ($CFL_x = 10$, $CFL_y = 1000$), which is within the stability boundary shown in Fig. 7.4. Uniformly good damping is observed at all wave modes including the purely axial modes along the x -axis of the stability diagram. Thus, the conditional stability of the LGS-4 scheme is not a serious impediment for two-dimensional problems.

The stability of the LGS-2 scheme depends upon the orientation of the high aspect ratio cells. If the line inversion is performed along the direction of the smallest cell dimension, then the stability is similar to the LGS-4 scheme, and the LGS-2 scheme is more efficient than the LGS-4 scheme because only half as many sweeps are performed. If the line inversion is performed in the direction of the largest cell dimension, then the stability is very poor (closely resembling the high aspect ratio LU results). For a general grid, the LGS-4 scheme is needed to account for both orientations of the grid cells, and gives better stability results than the two-sweep LGS scheme.

7.2.1.4 ADI Algorithm

The stability results for the upwind ADI algorithm (Eq. 7.6) are similar to the central-differenced results presented in Section V, and are not presented here. We point out, however, that the ADI-stability, like that of the other approximate methods, is markedly different from that of the direct scheme. In particular, there is an optimum value of CFL (usually around 5) for which the best convergence results are obtained. Furthermore, for high aspect ratio problems, choosing the time-step size according to the min-CFL definition (Eq. 5.2) enables good convergence.

7.2.2 Convergence Results—Euler Equations

To check the Euler stability results, a series of inviscid straight duct flows is computed with each of the upwind algorithms. The uniform grid consists of 41×41 points and the grid

aspect ratio is varied by reducing the width of the duct while keeping the length constant. This produces a uniform grid aspect ratio throughout the computational domain. For all cases the initial condition is taken to be a one-point, 10% perturbation of the uniform flow in the center of the domain and allows the Mach number and grid aspect ratio to be varied independently to assess their influence on convergence.

Figure 7.8 shows representative convergence results for the LGS-4 algorithm with I/III differencing at a Mach number of 0.001 and aspect ratios ranging from unity to 1000. These cases were run at a min-CFL of 20, which, in agreement with the stability results, was found to be optimum. At $AR=1$, values of min-CFL greater than 30 resulted in a low wave-number instability (as determined from contour plots of the residuals). At higher AR 's this instability was not encountered but the stability results indicate that the unstable region moves to lower wave-numbers as the aspect ratio increases. Not enough grid points were used in the present computations to encounter this instability mode.

The need for inviscid preconditioning at this low Mach number is also evident from Fig. 7.8. The $AR=100$ case was run with and without preconditioning. Without preconditioning, the convergence slows down by a factor of 10 compared to the case with preconditioning. All other AR results shown here were computed using inviscid preconditioning.

The change in slope of the convergence for $AR=100$ and 1000 appears to be an outflow boundary problem and doesn't occur for I/I differencing. Since the flattening occurs after the residuals reach 10^{-10} , it has little effect on overall convergence.

Figure 7.9 shows a summary plot of the number of iterations required to reach machine accuracy versus the grid aspect ratio for the LU, LGS-4 and ADI schemes, based on inviscid preconditioning and the min-CFL definition. It is evident that the LGS-4 and ADI schemes show convergence that is relatively independent of AR , while the convergence of the LU scheme suffers as the AR increases. This is in general agreement with the stability findings.

7.2.3 Navier-Stokes Stability Analysis

For Navier-Stokes computations, convergence is influenced by both convective and diffusive effects. The overall importance of the diffusive terms for a particular computational grid is characterized by a non-dimensional viscous time step, the von Neumann number ($VNN = \nu \Delta t / \Delta x^2$). The relative importance of convection and diffusion is given by the ratio, CFL/VNN , which corresponds to a cell Reynolds number. Cell Reynolds numbers ranging from less than unity to much greater than unity are typically encountered in practical high Reynolds number calculations. Boundary layer resolution requires strong grid stretching and very fine grids near the wall. These small grid sizes, coupled with the low velocities in the near wall region, give rise to small cell Reynolds numbers even when the overall computation represents a high Reynolds number flow. The larger grid sizes in the free stream (along with the higher velocities) result in large cell Reynolds numbers over most of the remainder of the field. This wide range of cell Reynolds numbers must be accounted for in time-step selection procedures for Navier-Stokes calculations.

In pure diffusion problems, the optimum local time-step selection should be based on the minimum von Neumann number in each grid cell, however, the Navier-Stokes equations

never approach this pure diffusion limit. Even for creeping flow, the presence of the continuity equation causes the inviscid modes to continue to play an important role in convergence behavior, and they become more important at moderate cell Reynolds numbers. Choosing the time step based only on the viscous characteristics of the problem causes these inviscid modes to languish. Accordingly, except for very large cell Reynolds numbers, the preconditioning parameter, ϵ , must be scaled so as to optimize the relevant inviscid and viscous modes simultaneously (Eqn. 4.6). The selection of ϵ_{vis} in the presence of high aspect ratio cells was discussed for central-difference schemes and the optimum min-CFL/max-VNN definition was given in Eqn. (5.4). As noted earlier, this definition equates the inviscid time scale in the stream-wise direction (represented by the minimum CFL) with the viscous time scale in the cross-stream direction (represented by the maximum VNN).

We again consider von Neumann stability results only for the LU and LGS schemes since the ADI results resemble those for the central-differenced system presented in Section V. All three upwind methods are, however, considered in the convergence studies.

7.2.3.1 Point Gauss-Seidel Scheme

Stability results of the point Gauss-Seidel scheme for high aspect ratio viscous problems are shown in Fig. 7.10 using the min-CFL/max-VNN viscous preconditioner (Eq. 5.4). Here, $AR=100$, $Re_{\Delta x} = 10$, $Re_{\Delta y} = 0.1$ and the preconditioning matrix maintains $CFL_x = 1 \times 10^6$ and $VNN_y = 1 \times 10^5$, which are the optimum values of these parameters for an aspect ratio of unity. For this higher aspect ratio, however, the stability diagram reveals that the purely axial modes (along the x -axis) are poorly damped. It is interesting that the LU scheme becomes unconditionally stable for viscous-dominated problems (as opposed to inviscid problems); however, the scheme still appears to be inefficient for the solution of high aspect ratio viscous problems. The stability analysis of the scalar heat conduction equation also reveals a similar stiffness of the axial modes when high aspect ratios are considered. Thus, the difficulty appears to arise from the basic nature of the point-Gauss-Seidel procedure and is not related to the definition of the preconditioning matrix.

7.2.3.2 Line Gauss-Seidel Scheme

The stability results for high aspect ratio viscous problems using the LGS-4 scheme are shown in Fig. 7.11. These results are for an $AR=100$ and $Re_{\Delta x}=10$, with $CFL_x=30$ and $VNN_y=10$. The amplification factors are again well conditioned over most of the domain, but are very stiff for purely axial modes. In an effort to check the sensitivity to the time-step definition, we checked the stability results for a min-CFL/min-VNN definition (Eqn. 5.3). This change in the time-step definition results in good damping over the entire wave-number spectrum, as shown in Fig. 7.12. For this case, $CFL_x=30$ and $VNN_y=10$, which is near the optimum values for these parameters. Note that the value of the viscous preconditioning parameter in Eq. (5.4) depends on the ratio of the CFL and VNN numbers and optimum performance is obtained for the LGS scheme when this ratio is maintained around 3. In contrast, the optimum CFL/VNN ratio is typically 0.3 for the ADI scheme and 10 for the LU scheme.

For viscous-dominated problems, the LGS scheme is also unconditionally stable (like the LU scheme); however, when larger time steps are used, the stability diagram ‘freezes’ and remains similar to the results in Figs. 7.11 and 7.12. This means that convergence behavior will not significantly improve for larger time-step sizes. Thus, viscous preconditioning is necessary even if much larger CFL’s and VNN’s are used. For instance, if the VNN is taken to be 1×10^6 , the CFL should be set to 3×10^6 so as to maintain the optimum CFL/VNN ratio of three. Although the results in Figs. 7.11 and 7.12 suggest that the LGS scheme should perform best with the min-CFL/min-VNN time-step, convergence studies presented in the following section show this is not true.

7.2.4 Convergence Examples – Navier-Stokes Equations

Fully-developed viscous duct flow is used to give an assessment of the Navier-Stokes stability findings and convergence of the LGS, LU and ADI schemes. Since the flow is fully-developed, no grid stretching is required and we use a uniform 41×41 mesh, where the grid aspect ratio is, again, varied by decreasing the width of the duct while keeping the length constant. The initial condition is a one-point, 10% perturbation of the fully-developed flow. The Mach number is 0.001 and $Re_{\Delta x} = 10$. As the grid aspect ratio increases, $Re_{\Delta x}$ remains constant but $Re_{\Delta y}$ decreases from 10 (at $AR=1$) to 0.01 (at $AR=1000$).

Figure 7.13 shows convergence results for the LGS-4 scheme at aspect ratios ranging from unity to 1000. Since the optimum ratio of CFL/VNN was found to be 3 for the LGS scheme, the min-CFL was set to 30 and the VNN was set to 10 for all cases. We were able to run the LGS scheme at much higher values of the CFL, however the parameter which controlled the convergence rate was the ratio of CFL/VNN which should be kept close to a value of 3, as indicated by stability. Both min-CFL/min-VNN and min-CFL/max-VNN definitions were used to assess their effectiveness. A number of different conclusions can be drawn from this figure. First, note that the convergence is extremely slow when no viscous preconditioning is used ($AR=100$ case). When the min-CFL/min-VNN viscous preconditioner is used, the convergence, surprisingly, is no better than the case without viscous preconditioning. Recall that the stability results suggest that the min-CFL/min-VNN viscous preconditioner should yield very good convergence (see Fig. 7.12). The reason for this behavior is not yet well understood. The remaining curves in Fig. 7.13 employ the min-CFL/max-VNN preconditioner and reveal good convergence for all AR.

Similar convergence tests show that the min-CFL/max-VNN preconditioner also performs best for the ADI scheme (both central-difference and upwind); however, the ratio of CFL/VNN should be kept at 0.3. For the LU scheme, the optimum ratio of CFL/VNN for this system was found to be around 10, in accordance with stability theory.

A summary plot of the number of iterations required to reach machine accuracy versus the grid aspect ratio for the viscous cases is shown in Figure 7.14. Here, the LU, LGS-4 and ADI results are included. The LU scheme suffers from an axial-mode stiffness regardless of whether viscous preconditioning is used or not. This is evident in Fig. 7.14 and is in agreement with the stability analyses. Both the LGS-4 and ADI schemes give good convergence as the aspect ratio is increased.

7.2.5 Timing Study

We also include an assessment of the relative computation times for the schemes under consideration. The cases were computed on a scalar machine (IBM RS/6000 model 370 workstation) and a vector machine (CRAY Y-MP). All schemes were vectorized. The computation times (sec/step/grid point) are listed in Table 7.1 for the ADI, LGS-2, LGS-4, and LU schemes and are normalized with respect to the computation time for the ADI scheme. The reference was chosen to be the ADI scheme because it required the shortest time per iteration per grid point of all the schemes. The LGS-4 scheme required the most time per step because of the four sweeps through the domain (as opposed to two sweeps for all the other schemes), and also due to the recursive nature of the Gauss-Seidel sweeping. The additional computation time per step was found to be offset by the rapid convergence in terms of iteration count. By basing the convergence of the algorithms on the CPU time (instead of iteration count), the LGS and ADI schemes were found to be competitive with each other, and consistently performed better than the LU scheme for the high AR computations.

CODE	SCALAR	VECTOR
ADI	1	1
LGS-2	1.08	1.15
LGS-4	1.63	1.54
LU	1.08	1.16

Table 7.1. Relative computation times for the ADI, LGS-2, LGS-4, and LU schemes. Times referenced to the ADI scheme: 4.0×10^{-4} sec/step/grid point (IBM 370), 4.8×10^{-5} sec/step/grid point (CRAY Y-MP)

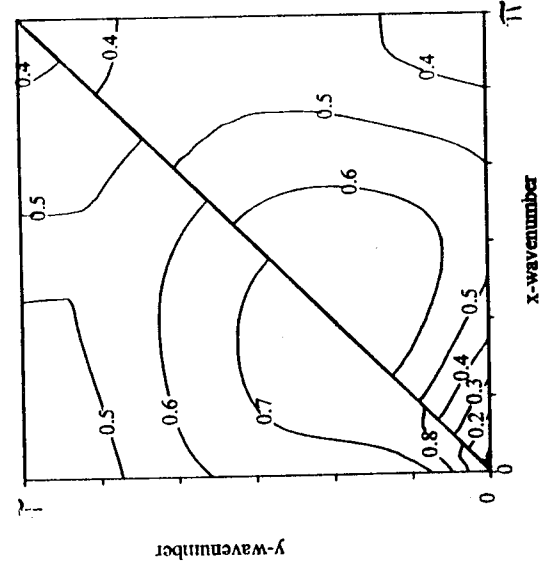


Fig. 7.1. Maximum eigenvalue of the amplification matrix for the Euler equations: Direct Inversion, first-order upwind LHS/third-order biased RHS (I/III), $M=0.7$, flow angle= 45° , $AR=1$. Upper triangle - CFL=10; lower triangle - CFL=1000.

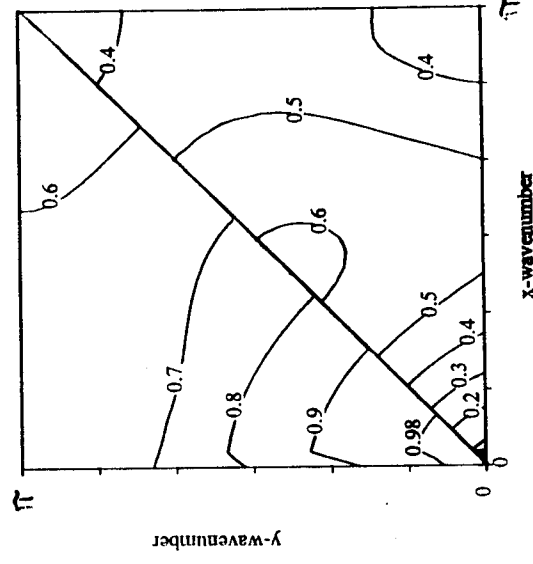


Fig. 7.2. Maximum eigenvalue of the amplification matrix for the Euler equations: Direct Inversion with inviscid preconditioning, (I/III), $M=0.001$, flow angle= 45° , $AR=1$, CFL=1000. Upper triangle - without preconditioning, lower triangle - with preconditioning.

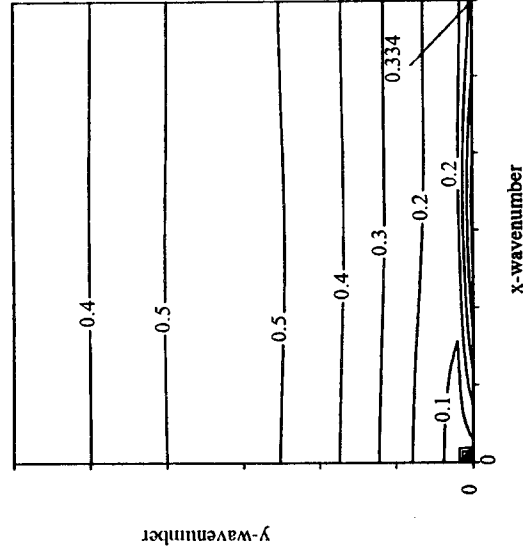


Fig. 7.3. Maximum eigenvalue of the amplification matrix for the Euler equations: Direct Inversion, (I/III), $M=0.7$, flow angle= 45° , $AR=100$, $CFL_x=1000$, $CFL_y=10^5$

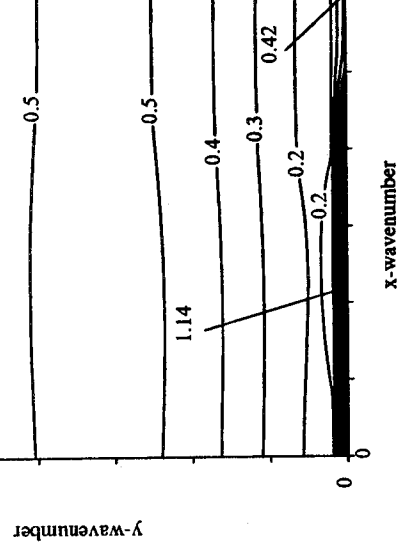


Fig. 7.5. Maximum eigenvalue of the amplification matrix for the Euler equations: LU, (I/III), $M=0.7$, flow angle= 45° , $AR=100$, $CFL_x=10$, $CFL_y=10^3$

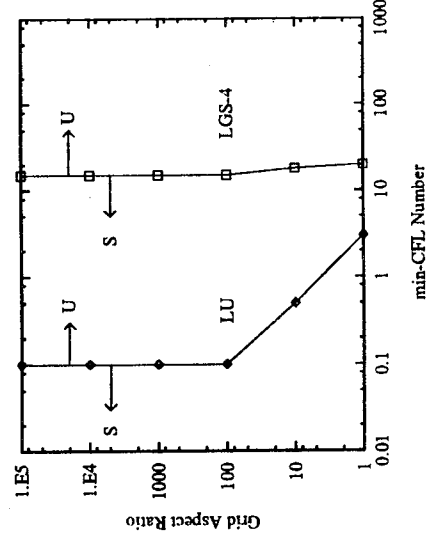


Fig. 7.4. Limiting stable minimum-CFL number versus grid aspect ratio for the LU and LGS-4 schemes. The region to the right of each curve (labeled "U") is a region of instability; the region to the left of each curve (labeled "S") is a region of stability

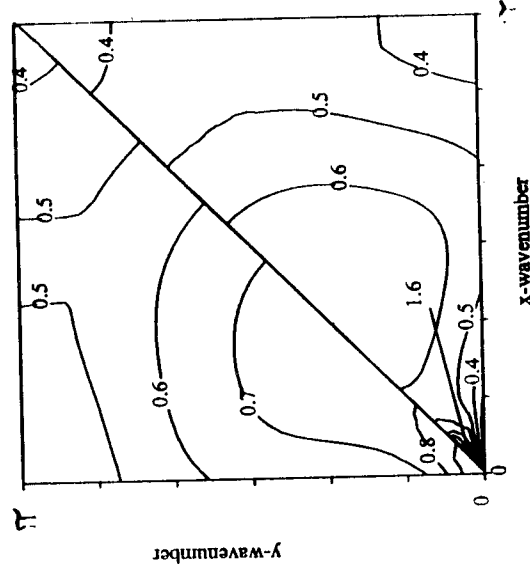


Fig. 7.6. Maximum eigenvalue of the amplification matrix for the Euler equations: LGS-4, (I/III), $M=0.7$, flow angle= 45° , $AR=1$. Upper triangle - $CFL=10$, lower triangle - $CFL=100$.

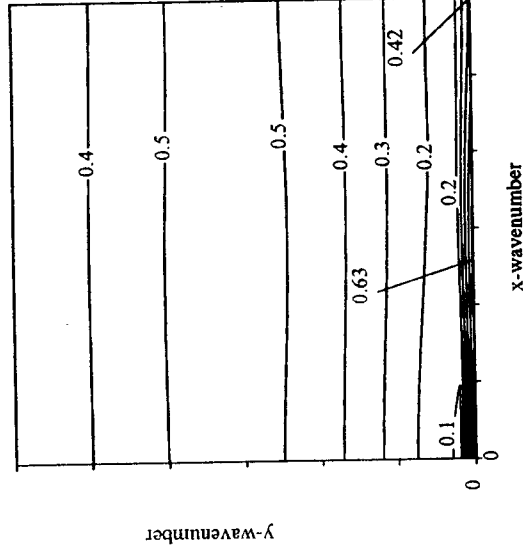


Fig. 7.7. Maximum eigenvalue of the amplification matrix for the Euler equations: LGS-4, (I/III), $M=0.7$, flow angle= 45° , $AR=100$, $CFL_x=10$, $CFL_y=10^3$

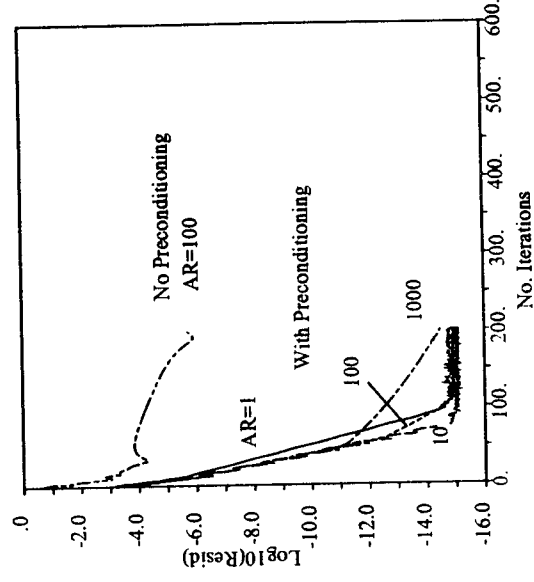


Fig. 7.8. Convergence results for inviscid straight duct flow at various AR 's using the LGS-4 scheme with I/III differencing. $M=0.001$, $\min-CFL=20$.

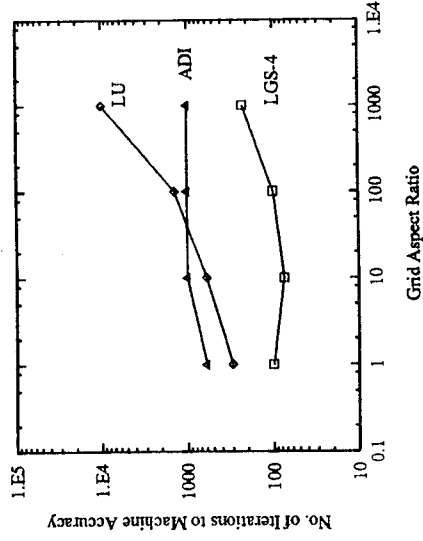


Fig. 7.9. Convergence to machine accuracy for the LU, LGS-4 and ADI schemes for the inviscid duct with aspect ratio as parameter.

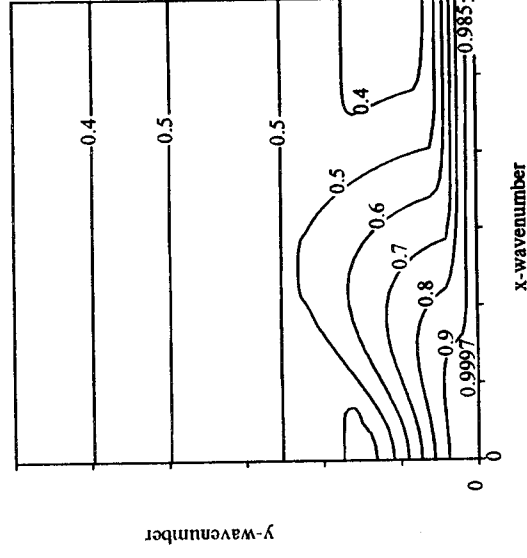


Fig. 7.11. Maximum eigenvalue of the amplification matrix for the Navier-Stokes equations with min-CFL/max-VNN preconditioner: LGS-4, (I/III), $M=0.001$, flow angle= 45° , $AR=100$, $Re_{\Delta x}=10$, $CFL_x=30$, $VNN_y=10$

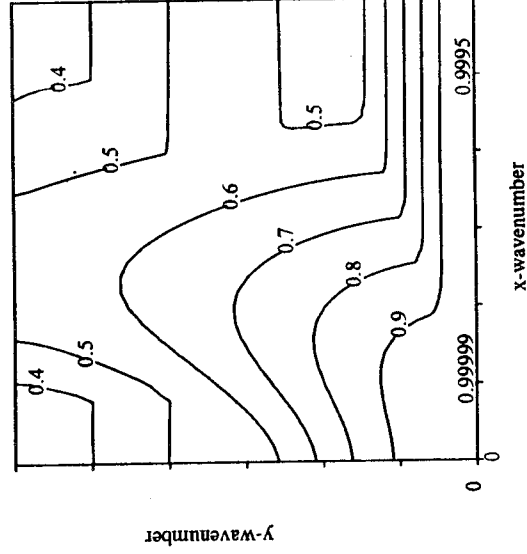


Fig. 7.10. Maximum eigenvalue of the amplification matrix for the Navier-Stokes equations with min-CFL/max-VNN preconditioner: LU, (I/III), $M=0.001$, flow angle= 45° , $AR=100$, $Re_{\Delta x}=10$, $CFL_x=10^6$, $VNN_y=10^5$

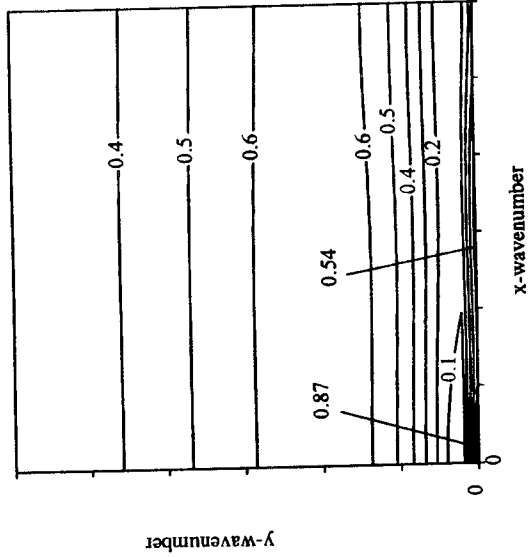


Fig. 7.12. Maximum eigenvalue of the amplification matrix for the Navier-Stokes equations with min-CFL/min-VNN preconditioner: LGS-4, (I/III), $M=0.001$, flow angle= 45° , $AR=100$, $Re_{\Delta x}=10$, $CFL_x=30$, $VNN_x=10$

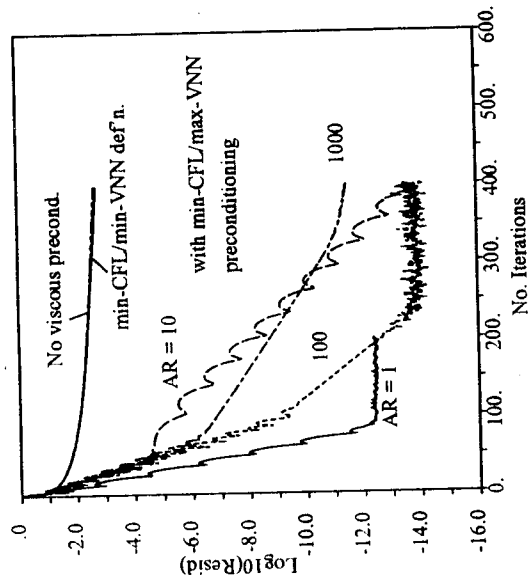


Fig. 7.13. Convergence results for fully-developed, viscous duct flow at various AR's using the LGS-4 scheme with I/III differencing. $M=0.001$, $Re_{\Delta x}=10$, $CFL=30$, $VNN=10$.

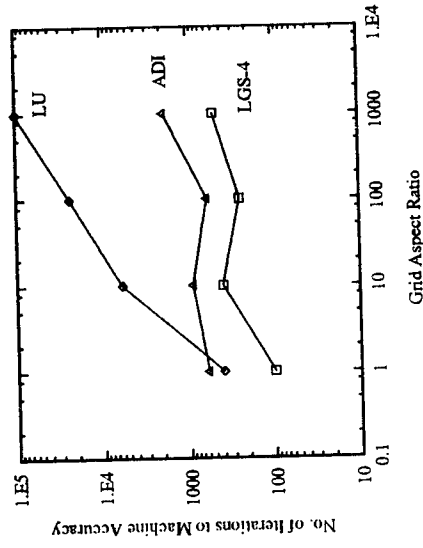


Fig. 7.14. Convergence to machine accuracy for the LU, LGS-4 and ADI schemes for the viscous duct with aspect ratio as parameter.

PRECONDITIONING FOR UNSTEADY COMPUTATIONS

8.1 Introduction

Both explicit and implicit algorithms are commonly used for unsteady computations. When applied as non-iterative, time-marching methods, these algorithms frequently lose temporal accuracy unless extremely small physical time-steps are used. This is particularly true for complex flowfields involving strong non-linear behaviour such as shock waves and combustion problems. Furthermore, stability criteria impose further limitations on physical time-step size especially for explicit algorithms. Similar time-step restrictions also exist for implicit algorithms in multi-dimensions because of errors associated with the approximate-inversion methods that are typically used. Particularly, in the presence of strong local grid-stretching or in low Mach number flows, such time-step restrictions may severely impair the usefulness of the algorithm. For these reasons, unsteady algorithms usually adopt some sort of iterative procedure at each physical time-level, which ensures temporal accuracy and, in the case of implicit schemes, also serves to eliminate linearization and approximate-factorization errors.

Two iterative approaches are in common use—namely, the approximate-Newton iterative procedure [42] or the dual-time-stepping approach [9, 40, 43, 44]. Both methods employ ‘inner’ or ‘sub’ iterations at each physical time-level and are, in fact, closely related. The dual-time-stepping method is the more general approach and the approximate-Newton method may be shown to be a subset of it. The dual-time method introduces a ‘pseudo’ time derivative (in addition to the physical time derivative) which acts as an agent to drive out errors in the physical transient and ensure convergence in the inner iterations. The potential advantage of the method lies in the fact that the pseudo-time derivatives may be chosen so as to optimize the convergence of the inner iterations. The selection of the pseudo-time derivatives may be achieved through the use of an appropriate preconditioning matrix that is tuned to the unsteady dual-time formulation.

Time-derivative preconditioning has been applied to unsteady computations previously in connection with the dual-time-stepping approach. However, in these instances, the same preconditioning matrix that was developed for steady-state computations was used. As we will demonstrate later, this is appropriate in some situations but ineffective in others. Accordingly, we carefully assess the numerical issues involved in solving unsteady flowfields. We use von Neumann stability theory to enhance our understanding of these issues and to guide the selection of the preconditioning matrix. We then use various test cases to confirm stability findings and to assess the capabilities of the dual-time scheme. We note that while our focus is on implicit algorithms, the formulation of the dual-time scheme is applicable to explicit algorithms as well.

Our specific interest lies in the solution of unsteady, low Mach number and compressible flows, such as diffusion flames. Because the physics of these flowfields are governed by the transport of vorticity, the characteristic speeds are the particle wave speeds. At low Mach numbers, there is a large disparity between the acoustic and particle time scales. Economical solution of these low-speed flows demands that the physical time steps be chosen to resolve the particle time scales and not the acoustic time scales. However, this requirement introduces serious numerical difficulties, which are discussed in detail later. Resolution of these difficulties within the framework of a compressible flow algorithm is the main focus.

The discussion is organized as follows. We start by discussing the numerical characteristics of approximate-Newton schemes and dual-time methods and their relationship to one another. We use linearized von Neumann stability analysis to explain convergence trends in the inner iterations and to identify the proper preconditioning matrix for the dual-time system. The preconditioned dual-time-stepping algorithm is then applied to a series of test cases—acoustic oscillations in a cylindrical chamber, the propagation of a Lamb vortex in a straight duct, and the spatial instability of a shear layer. We also employ analytical solutions to verify solution accuracy. In terms of convergence efficiency, we demonstrate that the preconditioned dual-time-stepping algorithm provides improved performance over a wide range of flow conditions and time scales.

8.2 Theoretical Development

8.2.1 Unsteady Euler Equations

The unsteady Euler equations in two dimensional Cartesian co-ordinates are given in their standard vector form:

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \quad (8.1)$$

Here, the flux vectors E and F have their traditional definitions for the equations in conservative form,

$$E = \begin{pmatrix} \rho u \\ (\rho u^2 + p) \\ \rho uv \\ (e + p)u \end{pmatrix}, \quad F = \begin{pmatrix} \rho v \\ \rho uv \\ (\rho v^2 + p) \\ (e + p)v \end{pmatrix}$$

and the primary dependent variable is given by $Q = (\rho, \rho u, \rho v, e)^T$.

In discretized space, the physical time scales associated with the wave speeds of the Euler equation system are given by the different CFL numbers. For example in the x direction, we have $CFL_u = u\Delta t/\Delta x$ and $CFL_{u\pm c} = (u \pm c)\Delta t/\Delta x$. These CFL numbers are of fundamental importance for the numerical solution of unsteady Euler problems. In particular, the physical time-step size should be selected based on whether acoustic processes or particle waves need to be resolved. For acoustics problems, it is generally necessary to maintain CFL_{u+c} of order unity (or less) to ensure temporal accuracy. Likewise, for unsteady flows governed by the particle wave speeds, CFL_u should be of order unity (or less). In practical situations, however, the time scales (or CFL numbers) are likely to vary

significantly within the solution domain because of local grid-stretching. In the following sections, we examine the role that the time scales play in the numerical solution of unsteady problems.

8.2.2 Approximate-Newton Method

In the approximate-Newton method, Eqn. (8.1) is solved iteratively at each physical time-level. The ‘sub’ or ‘inner’ iterations at each time-level serve to eliminate linearization and approximate-factorization errors and thereby ensure temporal accuracy. Here, the spatial derivatives in Eqn. (8.1) are treated using either second-order central differences or third-order upwind-biased differences. The temporal derivatives are treated using second-order accurate backward differences in time. The two-dimensional implicit operator resulting from the Euler implicit linearization of the fluxes is inverted approximately using one of several methods. For the central scheme, ADI approximate-factorization is used [13, 3, 4]. For the upwind-biased scheme, LU approximate-factorization (which is the same as symmetric point Gauss-Seidel relaxation) [36] or line-Gauss-Seidel (LGS) methods [38] are used. In the latter instance, we treat only the first-order flux terms implicitly in order to simplify the LHS matrix operator.

For simplicity, we present only the central-differenced ADI scheme:

$$\left[\frac{3}{2}I + \Delta t \frac{\partial A}{\partial x} \right] \frac{2}{3} \left[\frac{3}{2}I + \Delta t \frac{\partial B}{\partial y} \right] \Delta Q = -\Delta t \mathcal{R}^k \quad (8.2)$$

where the term $\mathcal{R}^k$ represents the residual error in the complete unsteady equation at inner iteration k ,

$$\mathcal{R} = \frac{3Q^k - 4Q^n + Q^{n-1}}{2\Delta t} + \frac{\partial E^k}{\partial x} + \frac{\partial F^k}{\partial y}$$

and $\Delta Q = Q^{k+1} - Q^k$. Equation (8.2) is the approximately factored expression that describes the inner iteration for the solution at time-level $n + 1$. At the first inner iteration, Q^0 is set equal to Q^n , the solution at time level n . Upon convergence, $\Delta Q \rightarrow 0$ and the unsteady Euler equations are clearly satisfied. The solution at the time level $n + 1$ is then given by $Q^{n+1} = Q^k$. Note also that if only one inner iteration is performed at each time level, the scheme reduces to the standard non-iterative time-marching method.

The convergence characteristics of the inner iterations are obviously dependent on the size of the physical time-step. In particular, the approximate-factorization error in Eqn. (8.2) is given by $[\frac{\Delta t^2}{\Delta x \Delta y} A B]$ and its magnitude is related to magnitudes of the CFL numbers. When the CFL numbers become much larger than unity, the factorization errors may dominate and preclude convergence. For acoustic problems and transonic and supersonic Mach numbers, the maximum physical CFL (i.e., CFL_{u+c}) is typically around unity and there is no difficulty with convergence. In fact, in the limit of small CFL's, Eqn. (8.2) becomes strongly sink-dominated (corresponding to the time-derivative term), which is extremely favorable to convergence. At these conditions, convergence to machine zero is typically attained in fewer than ten iterations. On the other hand, for low-speed problems,

when it is necessary to track particle waves, CFL_u is around unity while CFL_{u+c} can be several orders of magnitude larger. In such cases, we may anticipate convergence problems, as is indeed observed in practice.

Von Neumann stability analysis of the approximate-Newton method in Eqn. (8.2) is an effective method of analyzing the convergence behaviour. Figure 8.1, for example, shows the maximum amplification factor as a function of wave-number space for $M = 0.5$ and $CFL_{u+c} = 1$. All the other CFL numbers are therefore less than unity. The extremely small values of the amplification factor suggests excellent damping properties, an indication that Eqn. (8.2) is dominated by the sink-like temporal derivative at this condition.

Figure 8.2 shows the stability results for a low Mach number situation ($M = 0.001$). Taking advantage of the symmetry in the stability plots, we show results for two values of physical $CFL - CFL_{u+c} = 1$ (lower triangle) and $CFL_u = 1$ (upper triangle). The former choice of time-step is appropriate for resolving acoustic scales and since the corresponding $CFL_u = 1 \times 10^{-3}$, the factorization errors are small. Again, the excellent damping of the scheme indicates rapid convergence in the inner iterations. However, despite this, this choice of time-step is not suitable for tracking particle waves since tens of thousands of ‘outer’ (i.e., physical) time steps are necessary to advance the wavefront through the flowfield.

Stability results for $M = 0.001$ and $CFL_u = 1$ are also shown in Fig. 8.2 (upper triangle). This choice of physical time-step size is appropriate for resolving particle wave speeds. The amplification factors are, however, observed to be near unity over the entire wavenumber domain, suggesting very slow convergence rates in the inner iterations. This extremely stiff stability result is due to the large values of acoustic CFL ($CFL_{u+c} = 1000$), which means that the approximate factorization errors dominate over the physical terms in Eqn. (8.2).

Stability results (not shown) for the approximate-Newton method using upwind-based differencing and both the LU and LGS schemes show similar trends. In these schemes, the stiffness in the amplification factors (for low Mach number problems) is generally limited to the low wavenumber regions, while the inherent dissipation of these schemes provides adequate damping of the mid and high wave numbers. However, experience indicates that convergence difficulties are usually controlled by the stiff low wavenumbers. Thus, for low-speed problems such as unsteady mixing layers or diffusion flames, where the time scales are characterized by the particle velocity, one is faced with either taking a large number of outer steps with rapid inner-iteration convergence or taking few outer steps with slow inner-iteration convergence. In most instances, neither solution is practical. The dual-time stepping procedure, which is presented next, provides a way around this dilemma.

8.2.3 Dual-Time-Stepping Scheme

A more general procedure for obtaining iterative solutions to unsteady flowfields, and one which contains the approximate Newton method as a subset, is the dual-time approach,

wherein a 'pseudo' time-derivative is introduced in addition to the physical time-derivative in Eqn. (8.1):

$$\Gamma_p \frac{\partial Q_v}{\partial \tau} + \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \quad (8.3)$$

In this expression, $\Delta\tau$ is the pseudo-time step and Γ_p is the preconditioning matrix pre-multiplying the pseudo-time derivative. Note that the pseudo-time derivative is written in terms of our traditional vector of dependent variables $Q_v = (p, u, v, T)^T$, in accordance with the steady-state preconditioned system in the previous chapters. The second time-derivative is then responsible for the inner or sub-iterations at each physical time level. By judicious selection of the preconditioner for the second time derivative, convergence of the inner iterations may be optimized under different conditions. A one-parameter preconditioning matrix that has been successfully used for inviscid and viscous preconditioning of steady-state problems was discussed earlier and is given in Eqn. (2.3). For the unsteady dual-time scheme, we retain the form of this preconditioning matrix and look for the optimal specification of the parameter ϵ .

The dual time-stepping scheme may be written as follows:

$$\left[S + \Delta\tau \frac{\partial A_v}{\partial x} \right] S^{-1} \left[S + \Delta\tau \frac{\partial B_v}{\partial y} \right] \Delta Q_v = -\Delta\tau \mathcal{R}^k \quad (8.4)$$

where again central-differenced ADI is shown for simplicity. Note that Eqn. (8.4) has been derived by linearizing the governing equations in terms of the new dependent vector Q_v . Also, S is given by:

$$S = \Gamma_p + \frac{\Delta\tau}{\Delta t} \frac{3}{2} \Gamma_e$$

and Γ_e represents the Jacobian $\frac{\partial Q}{\partial Q_v}$ (defined in Eqn. 2.2). Once again, upon convergence of the inner (or pseudo) iterations, $\Delta Q_v \rightarrow 0$ and the residual $\mathcal{R}^k$ of the full unsteady equation is satisfied.

It is easy to see that when $\Gamma_p = \Gamma_e$ and $\Delta\tau \rightarrow \infty$, the dual-time scheme in Eqn. (8.4) becomes identical to the Newton scheme in Eqn. (8.2). Indeed, for unsteady transonic flow problems and acoustic problems, these are indeed optimum choices for these parameters. However, for unsteady low-speed problems involving slower characteristic time scales, we have seen that these choices are ineffective. Under such conditions, it is desirable to specify Γ_p and $\Delta\tau$ in a manner that guarantees optimum convergence. It is this inherent flexibility of the dual-time approach that makes it an attractive basis for the construction of improved unsteady algorithms.

One possible definition of Γ_p , which has been used by previous researchers [9, 44], is to take it to be the low Mach number steady-state preconditioning matrix (we will refer to this as Γ_s). In that case, the preconditioning parameter ϵ is given by Eqn. (2.5). The choice of M^2 renders the eigenvalues of the steady-state system well-conditioned at low Mach numbers, while the choice of unity gives the standard eigenvalues of the Euler system (i.e., the non-preconditioned system).

In order to evaluate, the potential of this preconditioner for unsteady problems, we consider results of von Neumann stability theory. Figure 8.3 shows the stability results for

this choice of Γ_p for $M = 0.001$ and the same physical CFL's as in Fig. 8.2. The pseudo-CFL ($\text{CFL}_\tau = \lambda \Delta \tau / \Delta x$, λ being the maximum eigenvalue of the preconditioned system) in both cases is taken to be 3, the optimum value for the ADI scheme. The lower triangle, which is for $\text{CFL}_{u+c} = 1$ and $\text{CFL}_u = 1 \times 10^{-3}$ is seen to be extremely stiff. In this case, all the physical CFL's are small and Eqn. (8.4) is sink-dominated (i.e., dominated by the physical time-derivative). Preconditioning the pseudo-time derivative according to the characteristics of the spatial derivatives is then inappropriate and leads to the poor stability map. In other words, the scaling of the pseudo-time-derivatives by the preconditioning parameter ϵ negates the favorable damping characteristics of the sink term. As noted earlier, the appropriate choice under these conditions is the Newton method, which involves setting $\Gamma_p = \Gamma_e$ and CFL_τ to infinity. This choice tunes the pseudo-time derivative to the sink term and maximizes the damping of the errors in the physical transient.

On the other hand, the steady preconditioning matrix does reasonably well when larger physical time step sizes are used as in the upper triangle, which is for $\text{CFL}_u = 1$ and $\text{CFL}_{u+c} = 1000$. Under these conditions, the spatial derivatives are more dominant, a consequence of the large acoustic CFL's. Thus, the selection of Γ_s is more appropriate. Note, however, that the low wave-number regions still show some residual stiffness (i.e., amplification factors are close to one). This suggests that convergence of the inner iterations may still be quite slow. If still larger physical time-steps are used, the stability improves further until it eventually resembles the steady-state stability diagram. The convergence of the inner iterations will then resemble steady-state convergence. In general, this limit is not of much importance for unsteady calculations.

From these results for low Mach numbers, it appears that different choices of the preconditioning matrix are required for the two limits of small and large physical time-steps. Furthermore, neither choice does particularly well for intermediate choices of time-step size (not shown). In problems involving strong grid-stretching, a wide range of physical CFL's are encountered. Thus, the proper selection of the preconditioning matrix is essential for the optimum solution of realistic flowfields. In the following section, we use linearized analysis to examine the characteristics of the dual-time scheme and use this information to derive an optimal preconditioning matrix.

8.2.4 Analysis of the Dual-Time Scheme

Von Neumann stability analysis of the one-dimensional version of Eqn. (8.3) is useful in enhancing understanding of the dual-time scheme because the amplification factors of the scheme may be derived analytically. The amplification matrix may be written as:

$$G = \left[\Gamma_p + \Gamma_e \frac{\Delta\tau}{\Delta t} + \frac{A_v \Delta\tau}{\Delta x} (iS_x) \right]^{-1} \Gamma_p \quad (8.5)$$

where $S_x = \sin(n\pi\Delta x/l)$. It is more convenient to write out the eigenvalues of G^{-1} rather than those of G . The corresponding amplification factors are then given by the reciprocals of these eigenvalues:

$$\begin{aligned} \lambda_1 &= 1 + \frac{\Delta\tau}{\Delta t} + \frac{u\Delta\tau}{\Delta x} iS_x \\ \lambda_{2,3} &= 1 + \frac{1}{2} \left[\left(\frac{\Delta\tau}{\Delta t} + \frac{u\Delta\tau}{\Delta x} iS_x \right) (1 + \epsilon) \pm \sqrt{\left(\frac{\Delta\tau}{\Delta t} + \frac{u\Delta\tau}{\Delta x} iS_x \right)^2 (1 - \epsilon)^2 - 4\epsilon\gamma RT \left(\frac{\Delta\tau S_x}{\Delta x} \right)^2} \right] \end{aligned} \quad (8.6)$$

Because these are the eigenvalues of the inverse matrix, the stability condition requires that the magnitudes of these complex eigenvalues be greater than unity and the larger the eigenvalues, the better is the damping. In Eqn. (8.6), ϵ is the preconditioning parameter that establishes the speed of the pseudo-acoustic waves in the preconditioned system. As noted earlier, $\epsilon = 1$ corresponds to the non-preconditioned case, while $\epsilon = M^2$ corresponds to the steady preconditioning.

Before we consider the definition of ϵ for the dual-time system, we briefly consider the eigenvalues in Eqn. (8.6) as we approach the limits of small and large time-step sizes. When all the physical CFL's are small (as they are for transonic and supersonic flows and problems involving acoustic waves), the eigenvalues become:

$$\lambda_1 = 1 + \frac{\Delta\tau}{\Delta t} \quad \text{and} \quad \lambda_{2,3} = 1 + \frac{1}{2} \left[\frac{\Delta\tau}{\Delta t} (1 + \epsilon) \pm \frac{\Delta\tau}{\Delta t} (1 - \epsilon) \right]$$

Here, the choice of $\epsilon = 1$ (no-preconditioning) renders the three eigenvalues equal. Note that all three eigenvalues show damping that is dependent upon the ratio of the pseudo to the physical time-step sizes. When $\Delta\tau$ is taken to be infinity and Δt is small, extremely rapid convergence will result. As noted earlier, this choice corresponds to the approximate-Newton method, which works well for such problems. On the other hand, when $\epsilon = M^2$ (steady preconditioning), one of the eigenvalues approaches unity as Mach number becomes low. This will correspond to poor convergence rates in the inner iterations. These conclusions are in agreement with the stability findings in Figs. 8.2 and 8.3.

Alternately, in the limit of large physical CFL's (the limit of steady-state), the eigenvalues become:

$$\lambda_1 = 1 + \frac{u\Delta\tau}{\Delta x}iS_x$$

$$\lambda_{2,3} = 1 + \frac{1}{2} \left[\frac{u\Delta\tau}{\Delta x}iS_x(1+\epsilon) \pm \sqrt{\left(\frac{u\Delta\tau}{\Delta x}iS_x\right)^2(1-\epsilon)^2 - 4\epsilon\gamma RT\left(\frac{\Delta\tau S_x}{\Delta x}\right)^2} \right]$$

It is clear that the magnitudes of these eigenvalues depend on the pseudo-CFL numbers, which is to be anticipated in the steady-state limit. At low Mach numbers, the acoustic-CFL number is much greater than the particle-CFL number. The second expression then reduces to $\lambda_{2,3} = \pm\sqrt{\epsilon\gamma RT}\frac{\Delta\tau}{\Delta x}iS_x$. Thus, λ_1 is governed by the particle-CFL number and $\lambda_{2,3}$ are controlled by the acoustic-CFL number. When $\epsilon = 1$ (no preconditioning) the disparity between the two sets of amplification factors causes the different parts of the equation to converge at vastly different rates. With approximate-inversion methods such as ADI, LU or LGS, this results in poor convergence rates. However, when $\epsilon = M^2$ (steady preconditioning), the three eigenvalues are of the same order of magnitude and the difficulty is resolved.

Examination of the eigenvalues in Eqn. (8.6) suggests the following definition of ϵ so as to maintain them of the same order of magnitude under all conditions:

$$\epsilon = \text{Min} \left[1, \frac{\left(\frac{\Delta x}{\Delta t S_x}\right)^2 + u^2}{c^2} \right]$$

The above expression depends on the wavenumber through the term S_x . Since in most problems, it is the low wavenumbers that are typically the hardest to damp out, we tune the ϵ definition so that it is optimum for the smallest wavenumber in the domain, i.e., $S_x = \pi\Delta x/l$ (i.e., $n = 1$).

Generalizing this definition of ϵ to two dimensions, we get:

$$\epsilon = \text{Min} \left[1, \frac{\left(\frac{l_x}{\Delta t \pi}\right)^2 + \left(\frac{l_y}{\Delta t \pi}\right)^2 + u^2 + v^2}{c^2} \right] \quad (8.7)$$

Here, l_x and l_y are characteristic length dimensions in the two co-ordinate directions. Note that in those regimes where the physical time-step size, Δt , is large, the preconditioning parameter ϵ becomes equal to M^2 which is the steady-state preconditioner. In the opposite limit, when Δt is very small, the preconditioning parameter becomes identically unity, and the no-preconditioning limit is recovered. In between, Eqn. (8.7) provides for a continuous transition between these two limits while maintaining good conditioning of the system.

We will refer to the preconditioning matrix, wherein the preconditioning parameter ϵ is defined by Eqn. (8.7), as the unsteady preconditioning matrix (Γ_u). In order to confirm the stability/convergence properties of this matrix for two-dimensional unsteady computations,

we consider the von Neumann stability results. Figure 8.4 shows these results for the central-differenced ADI scheme for $M = 0.001$ and two values of physical $\text{CFL} - \text{CFL}_{u+c} = 1$ (lower triangle) and $\text{CFL}_u = 1$ (upper triangle). In both cases, we take the pseudo-CFL (CFL_τ) to be three. In contrast to the schemes considered previously, the present scheme reveals good damping under both situations even though they correspond to widely different time scales. In the $\text{CFL}_u = 1$ situation, the approximate factorization error is evident over the mid wavenumbers causing the amplification factor to be around 0.9. However, as mentioned earlier, the convergence rate is primarily governed by the low wavenumbers, where this scheme shows better damping characteristics. Similar stability results for the upwind-biased LGS scheme are shown in Fig. 8.5. Here, the RHS fluxes are treated using third-order upwind-biased differences, while the LHS fluxes are treated using first-order upwind differences.

In the context of the dual-time scheme, it is important that the construction of the numerical fluxes be based on the preconditioned pseudo-time derivative and not on the physical time-derivative. The resulting diagonal dominance of the scheme allows for unconditional stability of the inner iterations and the pseudo-CFL (CFL_τ) may be picked to be very large (1×10^6 is used here). In Fig. 8.5, the lower triangle shows results for $\text{CFL}_{u+c} = 1$ and the upper triangle shows results for $\text{CFL}_u = 1$. Very good damping is again evident under both situations. Results without the unsteady preconditioning matrix suggest poor convergence behaviour as discussed earlier. Thus, the preconditioning matrix plays a dominant role in controlling the convergence characteristics of the upwind-based schemes as well. Comparison with Fig. 8.4 also indicates that the LGS scheme generally performs better than the ADI scheme. This is in accordance with experience from steady-state computations.

Stability studies (not shown here) also confirm the performance capabilities at intermediate time scales that may be encountered in problems involving local grid-stretching. Furthermore, stability results for the LU approximate-factorization scheme also reveal unconditional stability of the pseudo-iterations and good convergence behaviour at all Mach numbers and time scales. In the following sections, we examine the performance of the preconditioned dual-time scheme for a variety of representative test cases.

8.3 Computational Results

We consider several examples of unsteady flowfields such as acoustic oscillations, vortex propagation and unsteady mixing layers. We use these problems to verify the trends evident in the stability analyses of the approximate-Newton and dual-time schemes. For this purpose, we examine residual convergence in the inner iterations at a given physical time-level by carrying it out to machine zero. Clearly, this is not necessary in practice, where typically four orders of magnitude of convergence is sufficient.

These test cases also illustrate the capabilities of the preconditioned dual-time scheme both for acoustic wave processes as well as for low-speed flows characterized by the particle velocities. Performance comparisons are also made for different approximate-inversion methods such as ADI, LU and LGS. Finally, available analytical solutions are used to validate the numerical predictions.

8.3.1 Acoustic Oscillations in a Cylindrical Chamber

As a first example, we consider the problem of sustained acoustic oscillations in a cylindrical chamber. This problem involves tracking acoustic waves in the solution domain and therefore the physical time-step size should be based on the CFL_{u+c} . The conclusions from stability analysis indicate that both the approximate-Newton method and the dual-time scheme (using the unsteady preconditioning matrix) should provide extremely rapid convergence in the inner iterations. The test case thus provides a means of confirming these observations.

The acoustic oscillations are sustained within the chamber by enforcing a time-dependent inflow boundary condition of the form [45]:

$$(\rho u)' = \alpha p' + \beta u'$$

where $(\rho u)'$ is the fluctuation of the mass flow rate at the inlet of the chamber. Depending on the values of α and β , stable or unstable pressure oscillations may be sustained in the chamber. For the linearized, constant mean flow problem, analytical solutions may be readily obtained [45]. Here, we impose a specified mode solution (obtained analytically) as the initial condition and compute the ensuing temporal evolution of the disturbances. The temporal growth rates and frequencies are then compared with the analytical solution for validation. For further details of the physical problem and the analytical solution, the reader is referred to Ref. [45].

Figure 8.6 shows the convergence of the residuals in the inner iterations over several consecutive physical time-steps. For the case shown, the chamber Mach number is 0.3 and CFL_{u+c} is taken to be unity. Results are shown for both the approximate-Newton method as well as the dual-time preconditioned scheme. In both cases, central-differenced ADI is used. For the dual-time scheme, the pseudo-CFL is set to infinity which is the optimum value of this parameter under these conditions. Both schemes show rapid convergence in the inner iterations reaching machine zero in less than 20 iteration steps. Recall that for this choice of time-step, the unsteady equations are dominated by the temporal derivative which acts as a sink term in the inner iterations. If smaller values of time-step are used, even faster convergence is observed. These results thus validate the stability results and demonstrate that the dual-time scheme performs as well as the approximate-Newton method for acoustic wave problems.

Figure 8.7 shows a comparison of the numerical prediction with the analytical solution for the temporal variation of the pressure perturbation at a given location on the chamber wall. Excellent agreement is observed both in terms of the amplitude and the phase of the oscillations.

8.3.2 Propagation of Lamb Vortex

The next problem that we consider is the propagation of a Lamb vortex in a straight channel [46]. The initial velocity distribution corresponding to the vortex and is given by:

$$V_r(r, \theta) = 0, \quad V_\theta(r, \theta) = \Gamma \frac{[1 - \exp(-r^2/a^2)]}{r}$$

where Γ is the vortex strength and a is a characteristic radius. The initial specification of vorticity in the channel is shown in Fig. 8.8. The exact solution to the Euler equations specifies that the vortex travels downstream at the fluid particle velocity. For this study, we choose the channel Mach number to be 0.001. If the CFL_u is taken to be unity, the vortex would travel downstream one grid-line per physical time-step. Since the acoustic waves are not relevant to the vortex propagation, this is the desirable choice of time-step size. On the other hand, if CFL_{u+c} is taken to be unity, the vortex would propagate downstream by one grid cell per every one thousand iterations, which is not a very efficient time-marching procedure.

In the following study, we consider the convergence of the inner iterations for a single physical time step. As mentioned earlier, for diagnostic purposes, we carry out the inner iterations until machine zero is reached. In particular, we consider the convergence for different choices of the physical time-step size of the approximate-Newton method, the dual-time scheme with steady preconditioning (Γ_s) and the dual-time scheme with the unsteady preconditioning (Γ_u).

We start in Fig. 8.9 by choosing the time step size such that $CFL_{u+c} = 10$ and $CFL_u = 0.01$. At this choice of time-step, one hundred physical time steps are required to propagate the vortex through one grid cell. Convergence is shown for the three schemes (all with central-differenced ADI). For the dual-time scheme, $CFL_r = 5$. At this condition, the approximate-Newton iteration and the dual-time scheme with unsteady preconditioning converge almost as well taking about fifty iterations to reach machine zero. Note that the number of iteration steps has gone up compared to the result in Fig. 8.6. This slowdown is a consequence of the higher value of CFL_{u+c} which causes the convective derivatives to dominate for the acoustic wave modes. For the particle wave modes, as a result of the low value of CFL_u , the system continues to be dominated by the physical time-derivative. Figure 8.9 also shows that the steady preconditioner causes the convergence to completely flatten out, in accordance with stability findings.

In Fig. 8.10, we increase the choice of time-step to $CFL_{u+c} = 100$ and $CFL_u = 0.1$. Now, ten physical time-steps are necessary for the vortex to propagate through one grid cell. The convergence of the approximate-Newton method is now seen to flatten out, indicating that the approximate-factorization errors associated with the large CFL number has dominated the convergence process. At this condition, therefore, the Newton method becomes less useful. On the other hand, the dual-time scheme with unsteady preconditioning continues to converge quite efficiently, taking about 200 iterations to reach machine zero. Finally, the steady preconditioning is still observed to be extremely stiff.

Figure 8.11 shows the convergence of the three schemes with the time-step size given by $CFL_{u+c} = 1000$ and $CFL_u = 1$. As noted earlier, the vortex propagates downstream

in an optimum fashion—one grid line per physical time-step. At this choice of time-step, however, the approximate-Newton method becomes completely ineffective. In contrast, the dual-time scheme with unsteady preconditioning shows good convergence reaching machine zero in about 450 iterations. The steady-state preconditioner is only marginally better than in the previous cases.

The above results show that the properly preconditioned dual-time scheme maintains optimum convergence at all values of time-step. This is by itself a big advantage of the scheme since in complicated problems a broad range of time scales are likely to co-exist. Further, it is observed that the optimum convergence rate does slow down moderately with increasing time-step size (as indicated by stability theory in Fig. 8.4). This is due to the growing importance of the convective derivatives with increasing time-step size. For instance, comparing Figs. 8.9 and 8.11, we observe that the scheme requires ten times as many iterations in the latter case. However, the latter choice also requires a hundred times fewer physical time-steps than the former choice. Thus, for the Lamb vortex computation at a Mach number of 0.001, the dual-time scheme is more economical by a factor of ten. It should also be borne in mind that convergence to machine zero is generally not required. In practice, about 50-100 inner iterations would suffice and the associated CPU savings may be greater.

Finally, we consider the convergence results for the LU and LGS upwind-based schemes in Fig. 8.12. The results are for $CFL_{u+c} = 1000$ and $CFL_u = 1$. At this condition, the preconditioned dual-time scheme is again observed to outperform the approximate-Newton method by a factor of about ten. Furthermore, the preconditioned LU scheme is observed to be moderately faster than the ADI scheme (in Fig. 8.11), while the LGS scheme is substantially faster, reaching machine zero in about 75 iterations. These results are again in line with stability findings.

8.3.3 Spatial Instability of Mixing Layer

As a third and final demonstration, we consider spatially growing instabilities in a simple mixing layer. Figure 8.13 shows a schematic of the physical set-up. We consider a hyperbolic tangent shear layer whose mean profile is given by:

$$u(y) = \frac{u_1 + u_2}{2} + \frac{u_1 - u_2}{2} \tanh\left(\frac{y}{\delta}\right)$$

where u_1 and u_2 represent the velocities in the upper and lower portions of the shear layer, and δ represents the thickness [47]. The eigenfunction obtained from small perturbation analysis of the shear layer is used to furnish a 1% perturbation to the mean profile. This inlet condition is subsequently allowed to propagate through the computational domain.

Figure 8.14 shows the convergence in the dual-time for one physical time level. Again, the convergence is shown all the way to machine zero for the various preconditioners. The Mach number of the faster stream is 0.001, while that of the slower stream is 0.0005. The grid used here involves local stretching in the mixing region. Therefore, the physical CFL numbers vary in the flowfield (the physical Δt is held constant for temporal accuracy). The CFL_u varies from 0.1 to 1.0, while CFL_c varies from 100 to 1000.

From Fig. 8.14, it is evident that the unsteady preconditioner (in Eqn. 8.7) performs the best, reaching machine zero in about 600 iterations. The no-preconditioning case may be projected to take about 3500 iterations, a slowdown of a factor of six. Interestingly, the Newton iterations for this computation are divergent for this choice of physical CFL's (not shown). The steady preconditioner is again seen to converge extremely slowly. We note that even for a five order magnitude drop in the residuals, the unsteady preconditioner promises a CPU savings of a factor of at least five to six over the other methods.

Figure 8.15 shows the overall convergence over several physical time steps. With the current choice of physical Δt , each period of oscillation is resolved by about 30 physical time steps. On the other hand, if the approximate-Newton scheme were used, a much smaller physical Δt must be employed to ensure convergence in the inner iterations. Consequently, several thousand physical time steps would be needed to advance the solution through a single period of oscillation. Estimates of the projected savings in computer time range between a factor of five to ten depending on the level of convergence desired for the inner iterations.

Figure 8.16 shows the shear layer solution at a single instant in time. Axial and normal velocity contours are shown. Comparisons of the mode shapes with linear theory show good qualitative and quantitative agreement except near the exit boundary, where boundary reflections appear to impact the solution contours. Quantitative assessments of the accuracy of the predicted growth rates and frequencies are reported in Ref. [47].

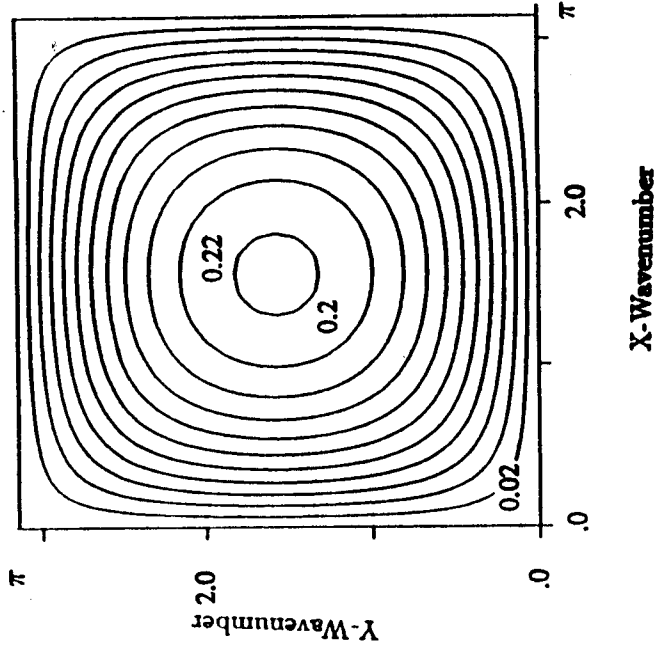


Fig. 8.1. Von Neumann stability map showing contours of the maximum amplification factor for the approximate-Newton scheme using central-differenced ADI. $M = 0.5$, $CF L_{u+c} = 1$.

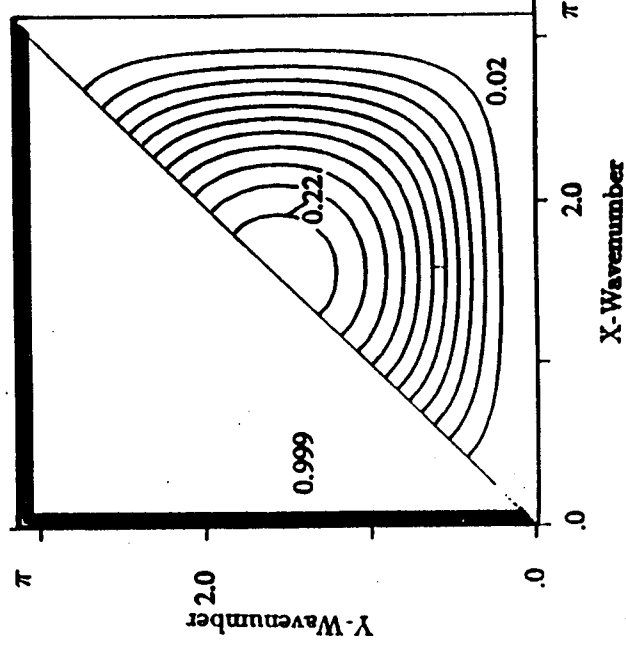


Fig. 8.2. Stability map for the approximate-Newton scheme (CD/ADI) for $M = 0.001$. Lower Triangle: $CF L_{u+c} = 1$. Upper Triangle: $CF L_u = 1$.

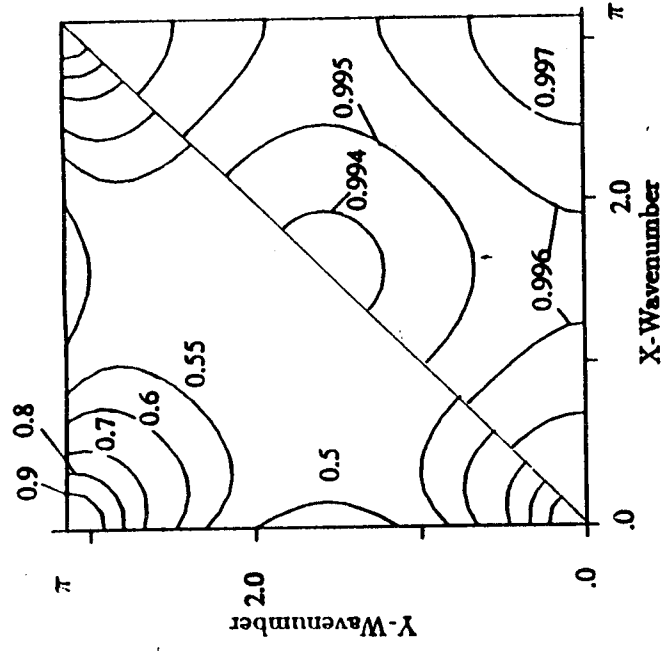


Fig. 8.3. Stability map for the dual-time scheme with steady preconditioning (CD/ADI) for $M = 0.001$ and $CFL_\tau = 3$. Lower Triangle: $CFL_{u+c} = 1$. Upper Triangle: $CFL_u = 1$.

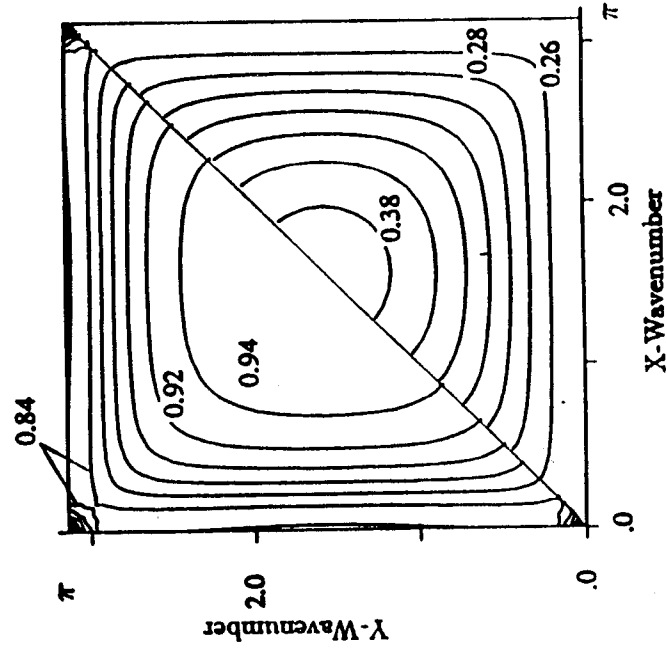


Fig. 8.4. Stability map for the dual-time scheme with unsteady preconditioning (CD/ADI) for $M = 0.001$ and $CFL_\tau = 3$. Lower Triangle: $CFL_{u+c} = 1$. Upper Triangle: $CFL_u = 1$.

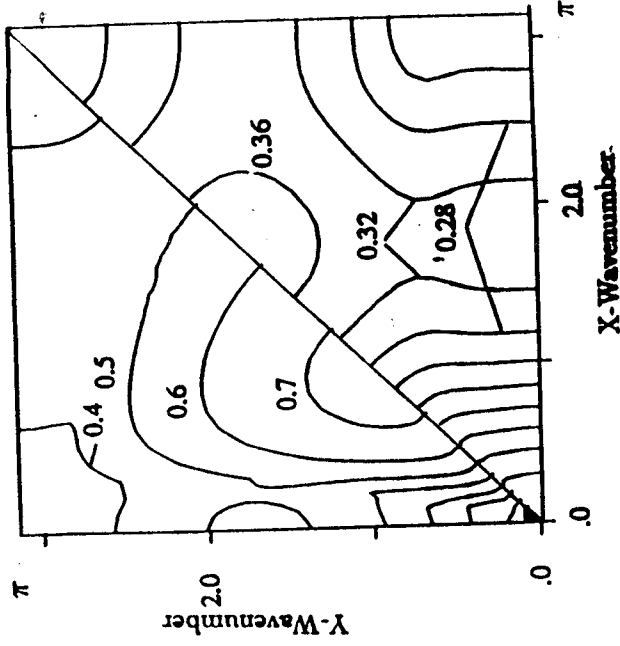


Fig. 8.5. Stability map for the dual-time scheme with unsteady preconditioning (UW/LGS) for $M = 0.001$ and $CFL_\tau = 3$. Lower Triangle: $CFL_{u+c} = 1$. Upper Triangle: $CFL_u = 1$.

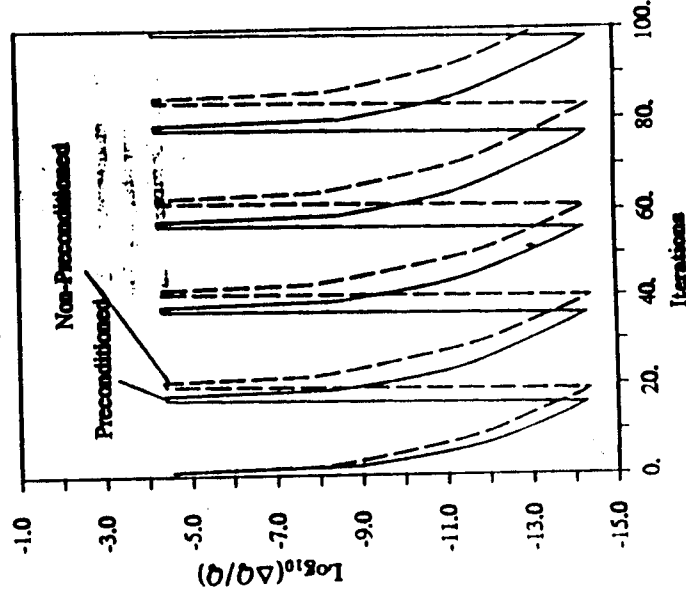


Fig. 8.6. Inner-iteration convergence shown for several consecutive physical time-levels for tracking acoustic oscillations in a cylindrical chamber. $M = 0.3$ and $CFL_{u+c} = 1$. Convergence is shown both for the approximate-Newton scheme and the preconditioned dual-time scheme.

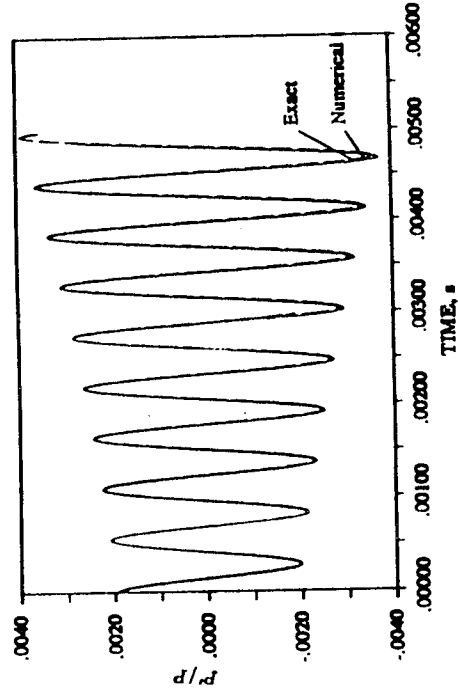


Fig. 8.7. Comparison of exact and numerical pressure oscillations on chamber wall location.

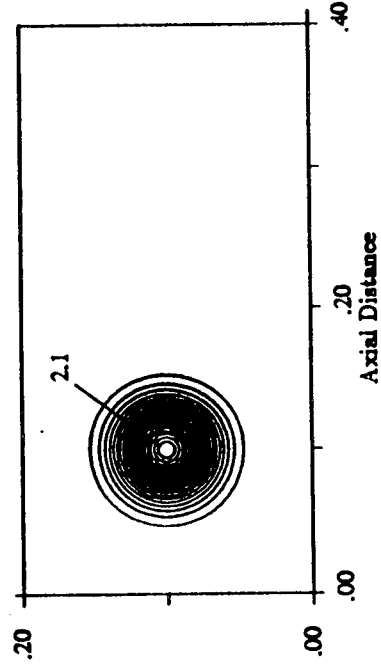


Fig. 8.8. Initial specification of vorticity for the propagation of Lamb vortex in a straight channel.

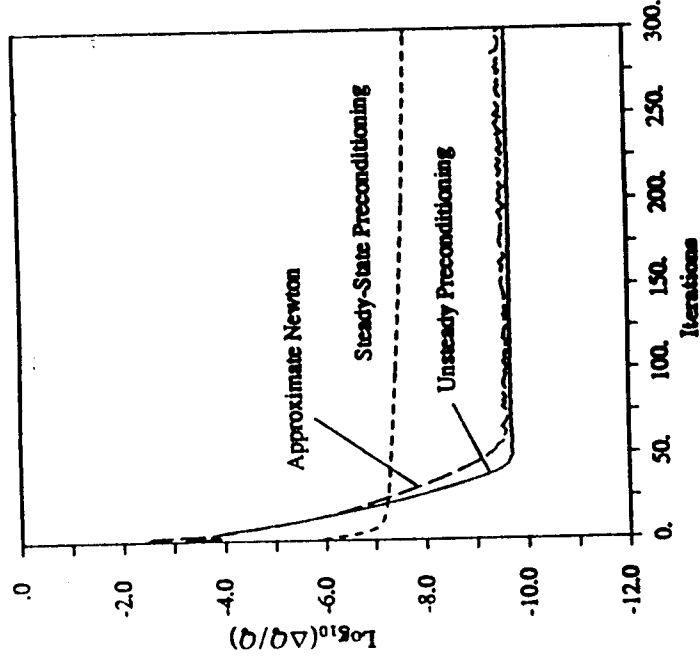


Fig. 8.9. Convergence rates of inner iterations for a single physical time-level in the propagation of Lamb vortex through a straight channel. $M = 0.001$, $CFL_{u+c} = 10$ and $CFL_u = 0.01$. Scheme is central-differenced ADI.

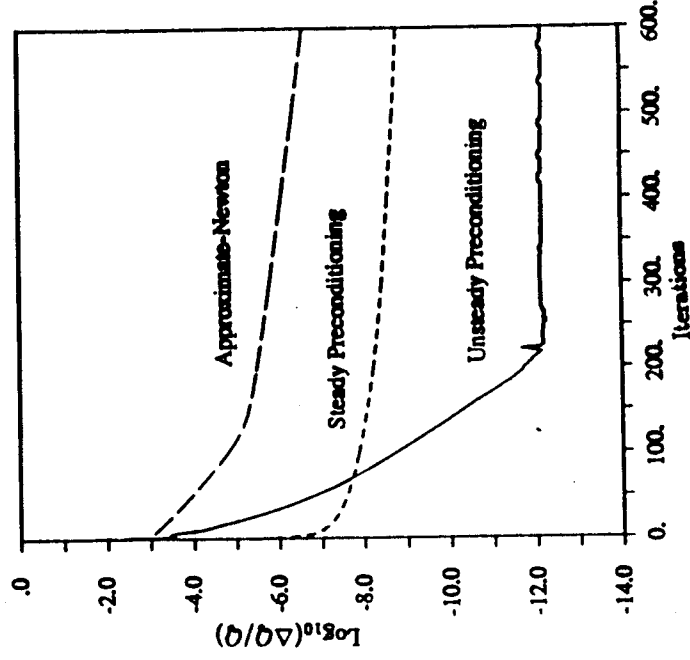


Fig. 8.10. Convergence rates of inner iterations for a single physical time-level in the propagation of Lamb vortex through a straight channel. $M = 0.001$, $CFL_{u+c} = 100$ and $CFL_u = 0.1$. Scheme is central-differenced ADI.

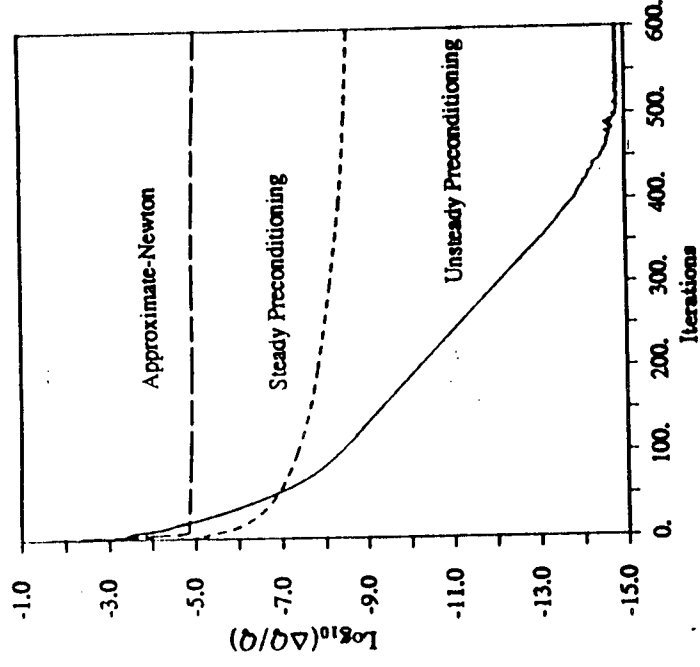


Fig. 8.11. Convergence rates of inner iterations for a single physical time-level in the propagation of Lamb vortex through a straight channel. $M = 0.001$, $CFL_{u+c} = 1000$ and $CFL_u = 1.0$. Scheme is central-differenced ADI.

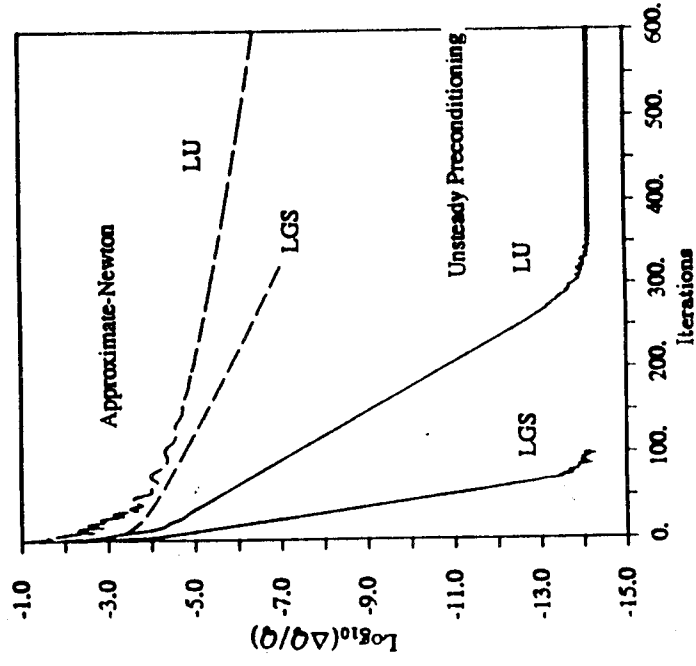


Fig. 8.12. Convergence rates of inner iterations for a single physical time-level in the propagation of Lamb vortex through a straight channel. $M = 0.001$, $CFL_{u+c} = 1000$ and $CFL_u = 1.0$. Third-order upwind-biased LU and LGS schemes.



Fig. 8.13. Schematic of unsteady mixing layer. $M_1 = 0.001$ and $M_2 = 0.0005$.

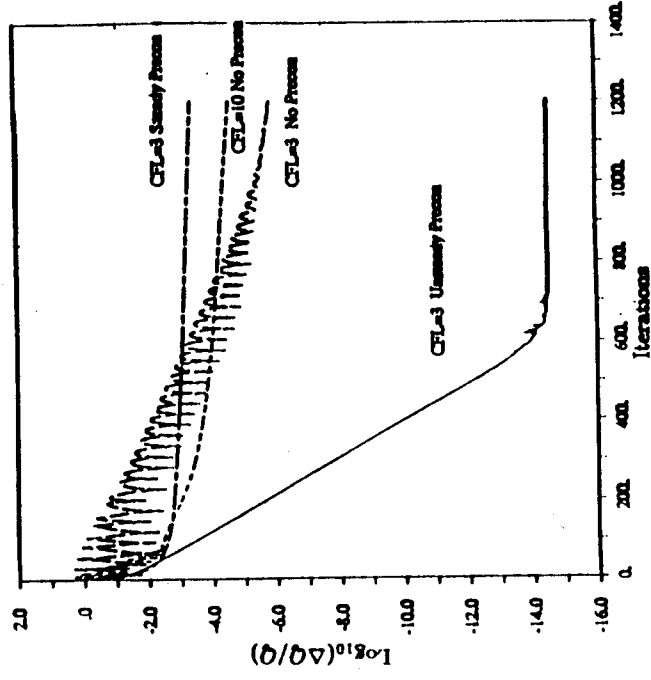


Fig. 8.14. Convergence rates of the inner iterations for a single physical time-level for the computation of a low-speed unsteady mixing layer (CD/ADI).

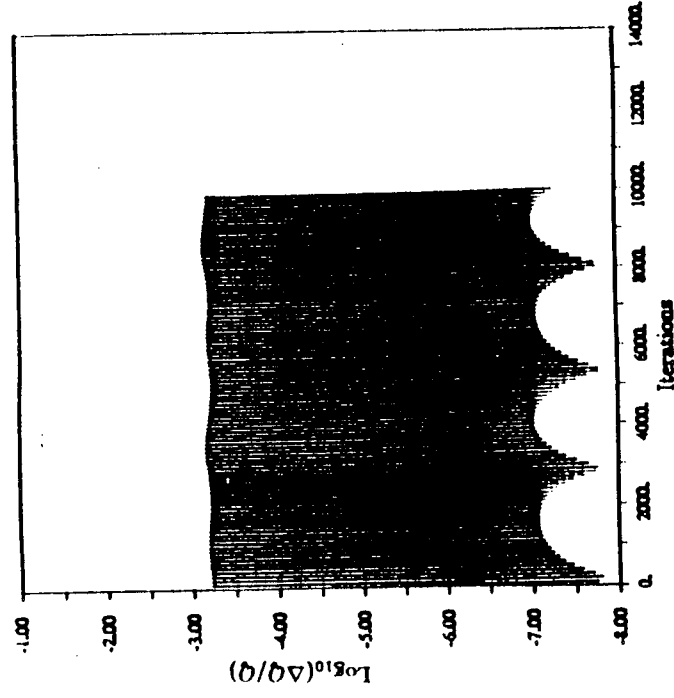


Fig. 8.15. Overall residual history for several consecutive physical time-steps in the computation of a low-speed unsteady mixing layer (CD/ADI).

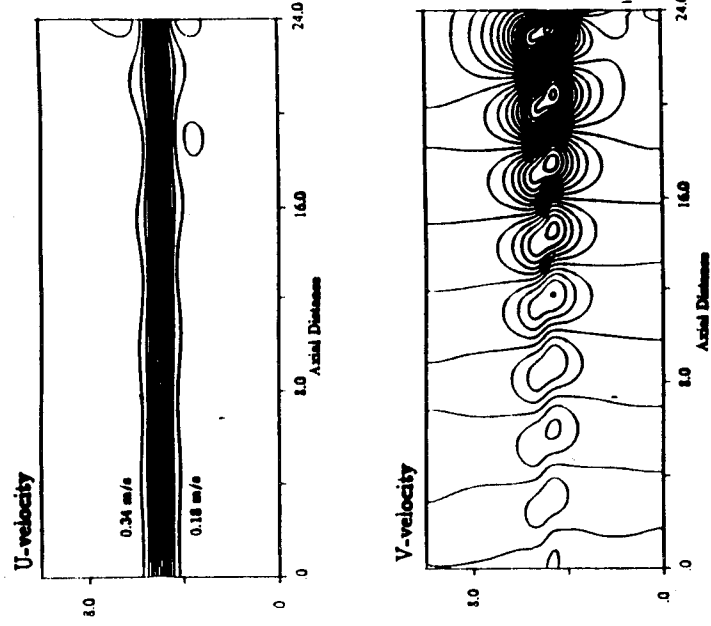


Fig. 8.16. Instantaneous U-velocity and V-velocity contours for unsteady mixing layer 0.01 s after start of computation.

GRID ASPECT RATIO EFFECTS ON 3D COMPUTATIONS

9.1 Introduction

High Reynolds number, turbulent flow computations necessarily utilize high aspect ratio grid cells in order to resolve high gradient regions. High aspect ratio cells (as well as high aspect ratio computational domains) invariably transform an efficient, robust code to a slowly converging, temperamental one. High grid aspect ratios have a strongly adverse effect on both upwind and centrally-differenced codes. Results presented in Sections V and VII demonstrated that for two-dimensional implicit schemes, this convergence degradation can be completely removed. In fact, it was demonstrated that the enhanced central-difference and upwind algorithms provide uniform convergence at all grid aspect ratios (to 10^6 and higher) for two-dimensional problems. These methods, however, deal only with two-dimensional algorithms. In three dimensions, grid aspect ratio problems become much more complex. For 2-D problems, the grid aspect ratio is simply defined as $AR = \frac{\Delta x}{\Delta y}$. For 3-D problems, there are now two grid aspect ratios to worry about, $\frac{\Delta x}{\Delta y}$ and $\frac{\Delta x}{\Delta z}$. These two aspect ratios define four different types of problems. To simplify the notation, we define 3-D aspect ratio problems of Type I as those where one cell dimension is small while the other two are relatively large. Similarly we define problems of Type II as those where two cell dimensions are small while the third is relatively large. Type III cells are defined as those where all three cell dimensions are non-commensurate in size (*e.g.* $\Delta x \ll \Delta y \ll \Delta z$). To complete the possibilities, Type 0 cells are defined as those where all three cell dimensions are commensurate and both $\frac{\Delta x}{\Delta y}$ and $\frac{\Delta x}{\Delta z}$ are of order one. The presence of these three generic kinds of high aspect ratio cells is responsible for the additional complexity involved in three-dimensional problems.

Besides these complexities encountered in 3-D high aspect ratio problems, the central-difference ADI algorithm is saddled with yet another short-coming that has always rendered it non-competitive in three dimensions. The characteristics of the ADI algorithm are such that the unconditional stability in two dimensions becomes only conditionally stable (and then only marginally so with maximum CFL's of near unity) in three dimensions. This conditional stability becomes particularly limiting at high aspect ratios as is shown herein.

In standard CFD algorithms, the definition of the local time step is typically based on the smallest grid dimension. The CFL number associated with this direction is thus specified at an optimum value, while the CFL numbers corresponding to the longer cell dimensions are forced to be very small. As a result, flow-field disturbances propagate slowly and are damped ineffectively in the longer cell direction, which results in poor convergence. In 2-D, the two factor ADI scheme, allows the local time step size to be specified on the basis of

the largest grid dimension. Thereby, the CFL numbers in both directions are maintained at optimum levels and the degradation in performance at high aspect ratios is avoided.

This attractive property of the ADI scheme unfortunately does not readily extend to 3-D problems. As noted above, the three-factor ADI scheme is conditionally stable. It displays reasonable convergence properties for Type I cells (which is an advantage over many other schemes), but not for Type II cells. In the latter case, the time step needs to be based on the two smaller dimensions resulting in a small CFL number in the longer cell dimension. The goal is to develop an algorithm that provides good convergence for all types of grid aspect ratios. For the sake of simplicity and also because this is often the case, we will take the axial cell dimension (x direction) as being long while the cross-plane dimensions (y and z) are both taken to be long (Type 0), one long and one short (Type I) or both short (Type II). Cases where all three cell sizes are non-commensurate (Type III) are also discussed.

The discussion is organized as follows. We begin by demonstrating the short-comings of the standard three-factor *ADI* scheme for various types of cell aspect ratios. We then formulate a two-factor *ADI* scheme that is unconditionally stable, albeit, one that is computationally inefficient. Our third step is to introduce a dual time stepping procedure designed to remove this inefficiency while taking advantage of the large cell aspect ratio success demonstrated for 2-D. In this dual time stepping approach, we introduce a second 'pseudo' time derivative that drives out the errors in the cross-plane, independent of the original 3-D time derivative. The original time derivative then becomes an agent to drive out streamwise disturbances while the second focuses on the cross-plane. The method is philosophically related to 3-D factorization algorithms with Newton-type iterations performed on the cross-plane [48]. In essence, the new scheme relies on the properties of two factor ADI, as opposed to those of the standard three-factor scheme.

An important aspect of the dual time stepping algorithm is the coupling between the two time-integration procedures. In particular, it is necessary to ensure that all parts of the equation are advanced to the same time-level. If formulated improperly, the scheme becomes ineffective for Type II cells. We show that this coupling may be enforced correctly by carefully selecting the initial condition for the pseudo-time iteration in the cross-plane. This then provides aspect ratio-independent convergence for cell aspect ratios of Type 0, I, II and III as is verified by computations with representative scalar equations and Euler stability findings. A stretched-grid Euler solution is presented for a converging-diverging nozzle case.

9.2 Three-Factor ADI Scheme

The 3-D preconditioned Euler equations are given by,

$$\Gamma \frac{\partial Q_v}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = 0 \quad (9.1)$$

The preconditioning matrix, Γ , is the same as that which was previously given in this report.

Applying Euler implicit discretization in time yields the following unfactored algorithm in delta form:

$$\left[\Gamma + \Delta t \frac{\partial A}{\partial x} + \Delta t \frac{\partial B}{\partial y} + \Delta t \frac{\partial C}{\partial z} \right] \Delta Q_v = -\Delta t \mathcal{R}^n \quad (9.2)$$

This fully implicit algorithm is unconditionally stable and Δt may be selected as large as necessary to obtain rapid convergence. However, the associated matrix operator is extremely expensive to invert. The conventional approach is to factorize the operator using any one of several approximate methods—such as ADI, LU or line Gauss Seidel (LGS). In such cases the time step size is limited by the factorization errors but good convergence is generally obtained for regular-sized grids.

The three-factor ADI scheme may be written as:

$$\left[\Gamma + \Delta t \frac{\partial A}{\partial x} \right] \Gamma^{-1} \left[\Gamma + \Delta t \frac{\partial B}{\partial y} \right] \Gamma^{-1} \left[\Gamma + \Delta t \frac{\partial C}{\partial z} \right] \Delta Q_v = -\Delta t \mathcal{R}^n \quad (9.3)$$

Stability results for the ADI scheme for regular and high aspect ratio grids are shown in Figs. 9.1-9.4. The three-factor ADI scheme is known to be conditionally stable, with a stability limit of $\text{CFL} \simeq 1.2$ for grids of aspect ratio unity ($\Delta x = \Delta y = \Delta z$). Figure 9.1 shows the amplification factor in the k_z, k_x plane for $k_y = 0, \pi/4, \pi/2$ for aspect ratio unity and $\text{CFL}=1$. For this condition, the scheme is stable and possesses good damping for all wave numbers. Note that the effect of the factorization error is quite apparent at the $\pi/2, \pi/2, \pi/2$ point, which is where the scheme eventually goes unstable for CFL's greater than 1.2.

We now examine the stability for high aspect ratio grids (namely, Type I and Type II cells). The stability results in the presence of Type I grids are shown in Fig. 9.2. The stability diagrams are shown only for the $k_y = \pi/2$ plane since this is the most restrictive. Here, $\Delta x/\Delta y = 1$ and $\Delta x/\Delta z = 10^6$. Grid aspect ratios of 1×10^6 are commonly encountered in turbulent flow computations. Accordingly, when the streamwise CFL (CFL_x) is unity, $\text{CFL}_y = 1$ and $\text{CFL}_z = 10^6$. Notice that now the scheme is unstable even when $\text{CFL}_x = 1$ and that this CFL needs to be lowered to about 0.5 to ensure stability. However, the CFL_z is now at a value of 5×10^5 . In spite of this large CFL in the z direction, the scheme is stable and well-conditioned over the entire wave number domain. Thus, the ADI scheme suffers only mildly in the presence of Type I cells and provides reasonable performance if the CFL number is appropriately chosen.

The stability results for Type II cells are shown in Figs. 9.3 and 9.4. Here, $\Delta x/\Delta y = 10^6$ and $\Delta x/\Delta z = 10^6$. In other words, the cell is very small in the cross-plane dimensions y and z while it is much longer in the streamwise co-ordinate x . Accordingly, when $\text{CFL}_x = 1$, $\text{CFL}_y = 10^6$ and $\text{CFL}_z = 10^6$. In Fig. 9.3, with $\text{CFL}_x = 1$, the scheme is seen to be unstable. Furthermore, the amplification factor is extremely stiff over most of the wave number domain. This is a result of the approximate factorization error term arising from the product of the two cross-stream derivatives (which is proportional to the product of CFL_y and CFL_z). In fact, even when CFL_x is lowered to 10^{-3} (Fig. 9.3b), the scheme, though stable, remains extremely stiff because of the high cross-plane CFL's ($\text{CFL}_y = \text{CFL}_z = 1000$). Figure 9.4 shows the stability for $\text{CFL}_x = 10^{-6}$ while $\text{CFL}_y = \text{CFL}_z = 1$. Now the stability diagram looks well conditioned over most of the wavenumber space. However, it approaches unity for purely streamwise modes (a consequence of the extremely low CFL_x), which is the

classical problem with high aspect ratio grids. Thus, the three-factor ADI scheme gives very poor convergence for grid cells of Type II no matter how the time-step is chosen.

9.3 Two Factor ADI Scheme for the 3-D Euler Equations

In order to preserve the favorable properties of the two factor ADI method, we factorize the 3-D equations into two factors, one for the streamwise direction and one for the cross plane:

$$\left[\Gamma + \Delta t_x \frac{\partial B}{\partial y} + \Delta t_x \frac{\partial C}{\partial z} \right] \Gamma^{-1} \left[\Gamma + \Delta t_x \frac{\partial A}{\partial x} \right] \Delta Q_v = -\Delta t_x \mathcal{R}^n \quad (9.4)$$

Note that we have written Δt_x in order to signify that the time step size is chosen based on CFL_x. The algorithm in Eqn. (9.4) is unconditionally stable (unlike the standard 3 factor ADI Eqn. (9.3)) and the time step can be optimized for all grid aspect ratios. Of course, the first operator is still a two-dimensional operator and is, therefore, expensive to invert. However, if we could devise an efficient method of inverting this operator exactly, the resulting algorithm would be very powerful for computing three-dimensional problems. This is the subject of the next section.

9.4 Formulation of Dual-Time-Stepping Scheme

One way of efficiently solving the two-dimensional cross-plane operator in Eqn. (9.4) is by iteratively solving for the first operator in Eqn. (9.4) through the introduction of a second pseudo-time derivative:

$$\Gamma' \frac{\partial \tilde{Q}}{\partial t_{yz}} + \left[\Gamma + \Delta t_x \frac{\partial B}{\partial y} + \Delta t_x \frac{\partial C}{\partial z} \right] \tilde{Q} = -\Delta t_x \mathcal{R}^n \quad (9.5)$$

where $\tilde{Q}$ represents the solution to the first operator in Eqn. (9.4) and t_{yz} represents a pseudo-time in the $y - z$ plane. The second time derivative is added to iterate on $\tilde{Q}$ until convergence is attained. Equation (9.5) may be written in the following approximately factored delta form:

$$\begin{aligned} \left[\Gamma' + \frac{\Delta t_{yz}}{\Delta t_x} \Gamma + \Delta t_{yz} \frac{\partial B^n}{\partial y} \right] \left[\Gamma' + \frac{\Delta t_{yz}}{\Delta t_x} \Gamma \right]^{-1} \left[\Gamma' + \frac{\Delta t_{yz}}{\Delta t_x} \Gamma + \Delta t_{yz} \frac{\partial C^n}{\partial z} \right] \Delta \tilde{Q} \\ = -\Delta t_{yz} \left[\mathcal{R}^n + \frac{1}{\Delta t_x} \Gamma \tilde{Q}^p + \frac{\partial B}{\partial y} \tilde{Q}^p + \frac{\partial C}{\partial z} \tilde{Q}^p \right] \end{aligned} \quad (9.6)$$

where $\Delta \tilde{Q} = \tilde{Q}^{p+1} - \tilde{Q}^p$. When $\Delta \tilde{Q}$ goes to zero, the right hand side residual ensures that the two-dimensional cross-plane operator in Eqn. (9.4) is solved exactly. The second time step size is represented as Δt_{yz} because it is selected on the basis of optimum convergence for the two-dimensional ADI operator in Eqn. (9.6) and differs from Δt_x by the aspect ratio of the grid. For high aspect ratio grids, $\Delta t_x > \Delta t_{yz}$. In particular, Δt_{yz} is chosen in a

fashion identical to that for two-dimensional high aspect ratio operators [41], thereby giving good convergence in the cross plane iterations no matter what the cross plane aspect ratio. Thus, Δt_{yz} is selected based on the $\text{Min}(\text{CFL}_y, \text{CFL}_z)$, while Δt_x is selected based on CFL_x .

A second important step in solving this cross-plane problem is the proper scaling of the preconditioning matrix in the cross-plane. For this reason, we have defined the preconditioning matrix Γ' rather than Γ in Eqn. (9.5). The matrix Γ' is selected to maximize the convergence rate of the cross-plane operator. Algebraically, Γ' is identical to Γ , but Γ' is based on the CFL's and the VNN's in the cross-plane, while Γ is based on considerations for the full three-dimensional operator. In other words, the parameter ϵ_{nv} is set equal to $M_{\text{ref}}'^2 = (v^2 + w^2)/c^2$ rather than the original reference Mach number given in Eqn. (9.2). The parameter ϵ_{vis} is likewise defined using the cell Reynolds numbers based only on the cross-plane.

A von Neumann stability analysis reveals that the attractive properties of the two-factor scheme in Eqn. (9.4) are preserved only if the initial value of $\tilde{Q}$ ($\tilde{Q}^o$) is properly specified. Otherwise, the time-advancement of the purely axial modes becomes extremely slow resulting in a corresponding stiffness of these modes.

The coupling between the two time integration procedures is clearly important to obtain the convergence properties of Eqn. (9.4). This coupling is manifest through the initial specification of the vector $\tilde{Q}$ for the cross-plane iteration. Initial results obtained by choosing $\tilde{Q}^o = 0$, which seems a reasonable choice, yields good results for Type I cells but not for Type II cells. Figs. 9.5-9.9 present these stability results.

Figure 9.5 shows the amplification factors for the dual-time ADI scheme when the cell aspect ratio is unity (i.e., $\Delta x = \Delta y = \Delta z$) and $\text{CFL} = 1$. Note that the dual time step scheme reduces to the three-factor ADI scheme when Δt_{yz} is taken to be very large compared to Δt_x and would thus have a stability map corresponding to the one shown in Fig. 9.1. In Fig. 9.5, $\Delta t_{yz} = \Delta t_x$ and only one dual time step was taken ($NP = 1$). Comparison of Fig. 9.5 with Fig. 9.1 shows that the damping properties of the two schemes are somewhat different. For the dual time step scheme, the low and high wave number modes are damped less while the mid-wavenumber modes are better damped. It should be pointed out that, if a larger number of inner iterations is performed, the dual time step scheme becomes unconditionally stable and its damping properties become similar to the two-factor ADI scheme. The results in Fig. 9.5 show that this is in general not necessary since one inner iteration is sufficient to obtain a stable scheme with good damping properties.

Figures 9.6 and 9.7 show the amplification factors for Type I cells with an aspect ratio of 10^4 ($\frac{\Delta x}{\Delta y} = 1, \frac{\Delta x}{\Delta z} = 10^4$). Figure 9.6 shows the results for the two-factor ADI (i.e., with the 'inner' iterations fully converged), while Fig. 9.7 shows the results for only 10 inner iterations. In both cases, the outer time-step Δt_x is chosen so that $\text{CFL}_x = 1$ while the inner time-step Δt_{yz} is chosen so that $\text{CFL}_y = 1$. Comparing the two results, it is evident that the amplification factors with only 10 inner iterations are essentially the same as those for the two factor scheme.

Figures 9.8 and 9.9 show amplification factors for Type II cells with an aspect ratio of 10^4 . Here, $\frac{\Delta x}{\Delta y} = \frac{\Delta x}{\Delta z} = 10^4$. Again, Fig. 9.8 is for the two-factor ADI (fully converged

inner iterations), while Fig. 9.9 is for the dual time step scheme with 10 inner iterations. Interestingly, the stability plots are quite similar over much of the wavenumber domain except in the low and high y and z wavenumber regions, where it becomes very stiff. The low wave number region corresponds to purely streamwise modes and will clearly lead to convergence degradation. It is also interesting to note that if the number of inner iterations were increased, this stiff region would gradually become well-conditioned. For this case, it takes over 10^4 iterations for Fig. 9.9 to exactly resemble Fig. 9.8, which is clearly too many to present any real savings.

At first glance, the results in Fig. 9.9 seem puzzling. The cross-stream iterations should not impact the streamwise modes and yet they do. After reflection, however, it becomes obvious that the problem lies in the time-integration of the axial modes in Eqn. (9.4). It is necessary to insure that these modes are advanced explicitly in time through a time interval of Δt_x in inverting the first operator of Eqn. (9.4). For an aspect ratio of 10^4 , Δt_{yz} is much much less than Δt_x , and it takes several thousand iterations for this time advancement to take place when the initial value of $\tilde{Q}$ is taken as zero. The problem is remedied easily by first advancing the operator through a time interval of Δt_x to give $\tilde{Q}^0$ and *then* introducing the second time derivative for the cross-plane iterations. The cross-plane iterations may then be interpreted as successive attempts to refine the value of $\tilde{Q}$. Accordingly, the initial condition for the cross plane operator is obtained by solving the following equation:

$$\left[\Gamma + \Delta t_x \frac{\partial B}{\partial y} \right] \Gamma^{-1} \left[\Gamma + \Delta t_x \frac{\partial C}{\partial z} \right] \tilde{Q}^0 = -\Delta t_x \mathcal{R}^n$$

Note that in this case, the three-factor ADI scheme is recovered if *no* inner iterations are made.

Figure 9.10 shows the amplification factors for the dual-time ADI scheme with this choice of the starting vector $\tilde{Q}^0$, when the cell aspect ratio is unity (i.e., $\Delta x = \Delta y = \Delta z$) and $\text{CFL} = 1$. Here only one dual time step is taken, and the amplification factors show improved damping over the three-factor ADI scheme for the mid y wavenumbers. The damping at $k_y = 0$ is identical to the three-factor ADI scheme.

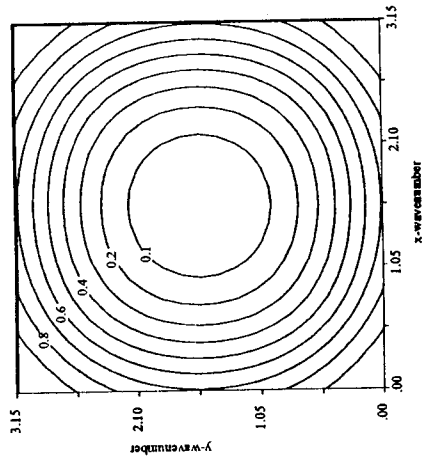
The stability characteristics for a Type I high aspect ratio cell ($\frac{\Delta x}{\Delta y} = 1$, $\frac{\Delta x}{\Delta z} = 10^4$) with 10 dual time steps and a $\text{CFL}_x = 1$ are shown in Fig. 9.11. As before, note that the two-factor ADI stability characteristics are recovered with less than 10 dual time steps.

The results for the Type II cells show the advantage to this selection of $\tilde{Q}^0$. Figure 9.12 shows the amplification factors for Type II cells ($\frac{\Delta x}{\Delta y} = \frac{\Delta x}{\Delta z} = 10^4$) with 10 dual time steps and $\text{CFL}_x = 1$. Comparing Fig. 9.12 with Fig. 9.8 reveals that the complete two-factor ADI stability characteristics can be recovered in about 10 dual time steps, when the proper choice of $\tilde{Q}^0$ is taken. Thus, the stiffness associated with the streamwise modes, as well as the high y and z wavenumber modes, has been alleviated through simply making the correct choice of the starting vector for the dual time iterations.

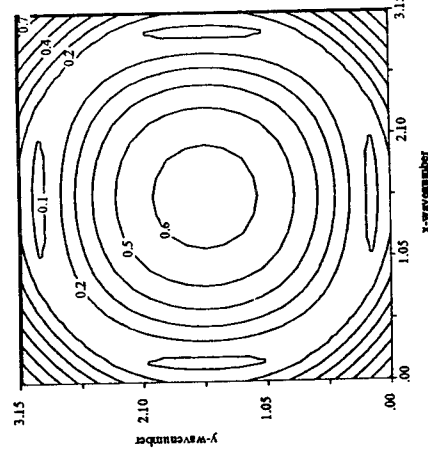
9.5 Computational Results

The dual time-stepping algorithm has been used to compute the choked flow in a rectangular cross-section, converging-diverging nozzle. The grid consists of $31 \times 31 \times 31$ points. Two cases have been computed. The first utilizes a uniform grid, and the second employs a stretched grid where the maximum Type II grid aspect ratio is approximately 40. Figure 9.13 shows the two grid systems where the ratio of throat area to inlet area is 0.36.

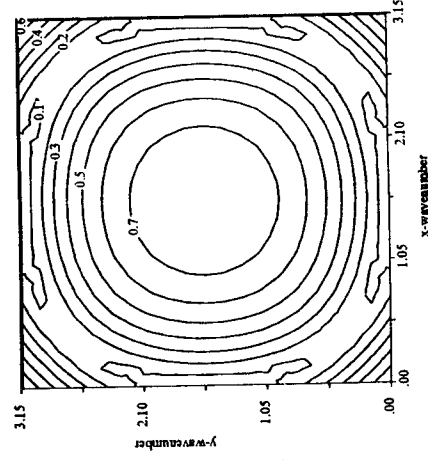
Convergence for the uniform grid case is presented in Figure 9.14, where the dual time scheme is compared with the standard three-factor ADI scheme. In this case, the standard ADI scheme converges reasonably well and the dual time scheme provides only slight improvement. Figure 9.15 shows convergence for the two algorithms on the stretched grid case. Here the dual time scheme shows a substantial improvement over the standard ADI scheme. In fact, the dual time scheme converges in the same number of iterations for both the uniform and stretched grid cases.



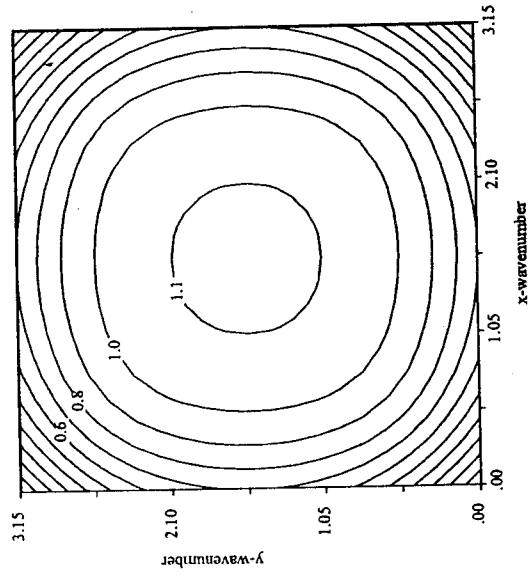
$$\omega_z = 0$$



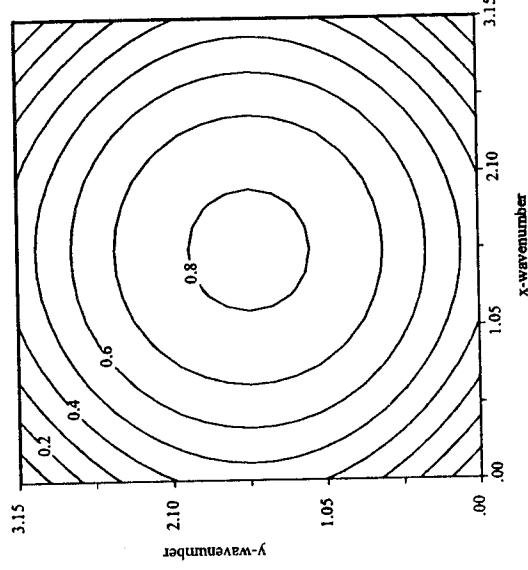
$$\omega_z = \frac{\pi}{4}$$



$$\omega_z = \frac{\pi}{2}$$



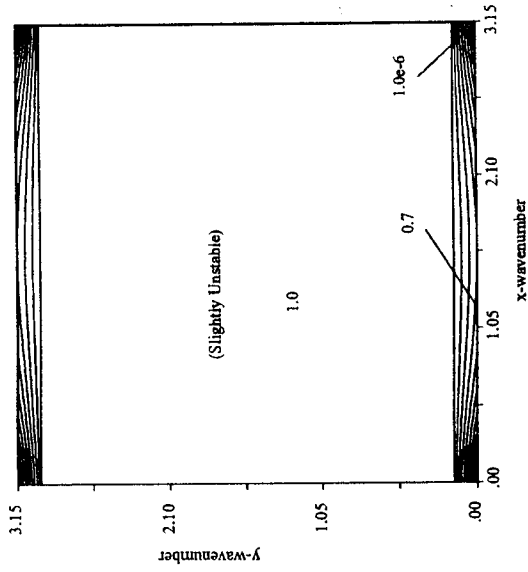
$$\omega_z = \frac{\pi}{2}, \text{ CFL}_x = 1$$



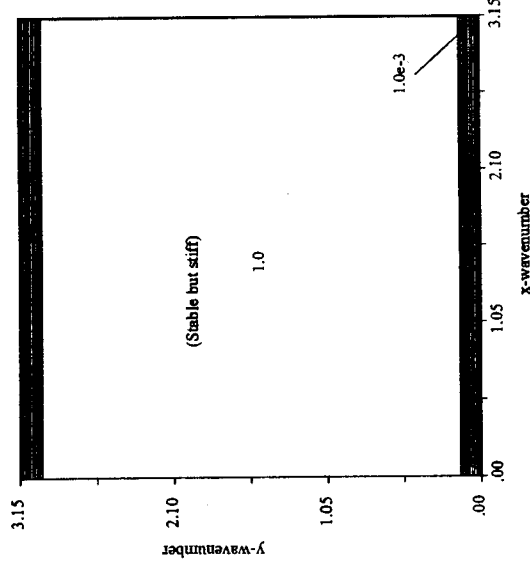
$$\omega_z = \frac{\pi}{2}, \text{ CFL}_x = 0.5$$

Fig. 9.1. Amplification factors for the three-factor ADI scheme applied to the scalar analog equation. $\Delta x = \Delta y = \Delta z$, $140 \text{ CFL}_x = 1$

Fig. 9.2. Amplification factors for the three-factor ADI scheme applied to the scalar analog equation. TYPE I cells $\frac{\Delta x}{\Delta y} = 1$, $\frac{\Delta x}{\Delta z} = 10^6$

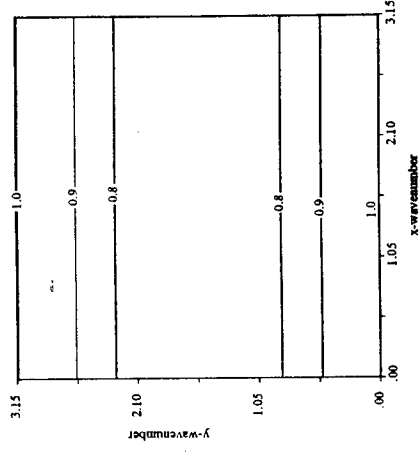


$$\omega_z = \frac{\pi}{2}, \text{ CFL}_x = 1$$

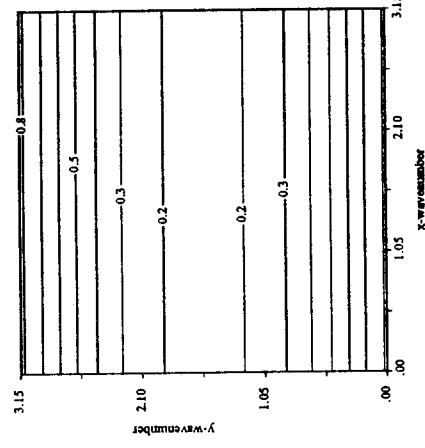


$$\omega_z = \frac{\pi}{2}, \text{ CFL}_x = 10^{-3}$$

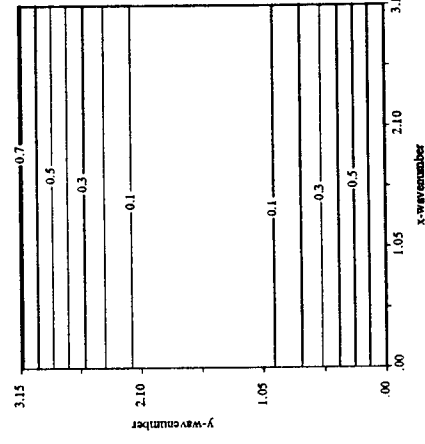
Fig. 9.3. Amplification factors for the three-factor ADI scheme applied to the scalar analog equation. TYPE II cells $\frac{\Delta x}{\Delta y} = 10^6$, $\frac{\Delta x}{\Delta z} = 10^6$



$$\omega_z = 0$$



$$\omega_z = \frac{\pi}{4}$$



$$\omega_z = \frac{\pi}{2}$$

Fig. 9.4. Amplification factors for the three-factor ADI scheme applied to the 141 scalar analog equation. TYPE II cells $\frac{\Delta x}{\Delta y} = 10^6$, $\frac{\Delta x}{\Delta z} = 10^6$, $\text{CFL}_x = 10^{-6}$

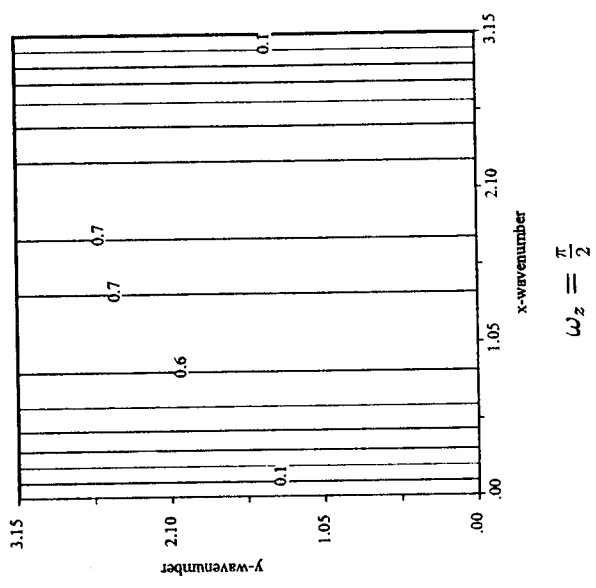
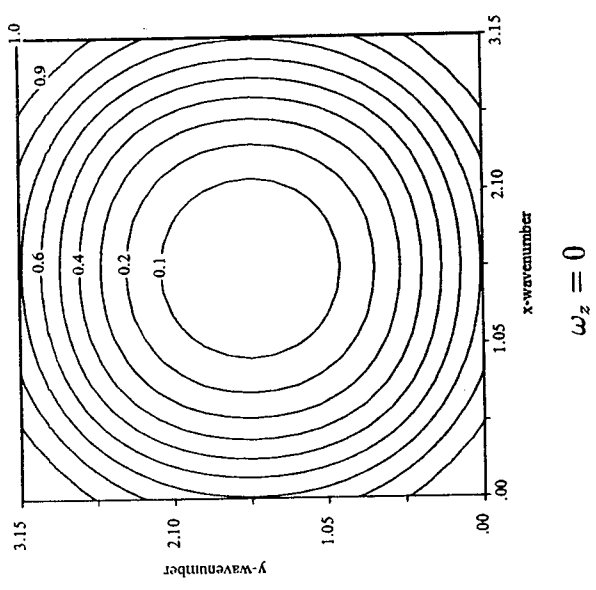
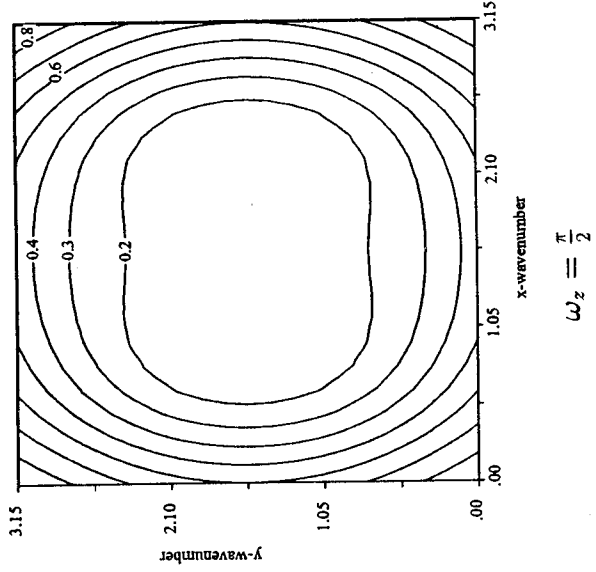
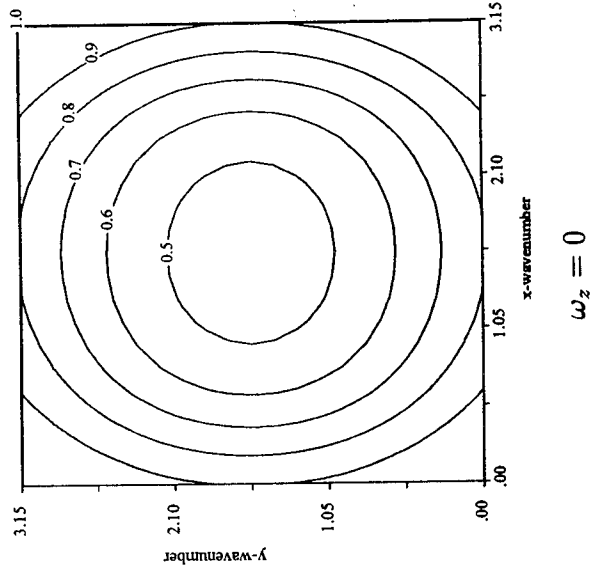


Fig. 9.5. Amplification factor for the dual time step ADI scheme with $\hat{Q}^o = 0$, $\Delta x = \Delta y = \Delta z$, $CFL=1$, $\frac{\Delta t_{yz}}{\Delta t_x} = 1$, $NP = 1$

Fig. 9.6. Amplification factor for the two-factor ADI scheme, TYPE I, $\frac{\Delta y}{\Delta z} = \frac{\Delta x}{\Delta z} = 10^4$, $CFL_z=1$

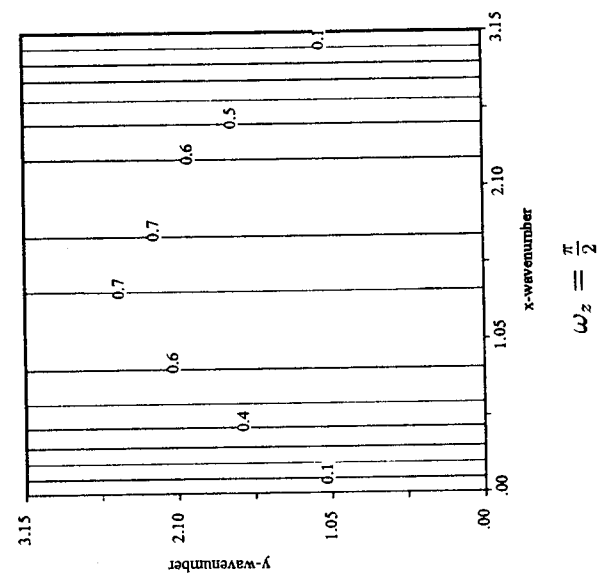
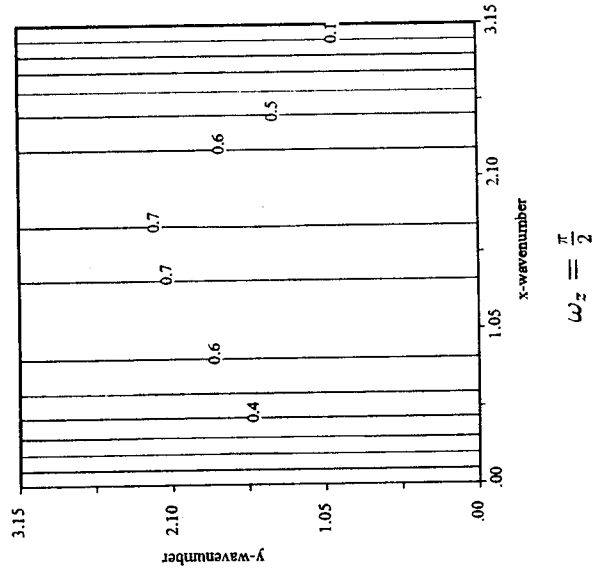
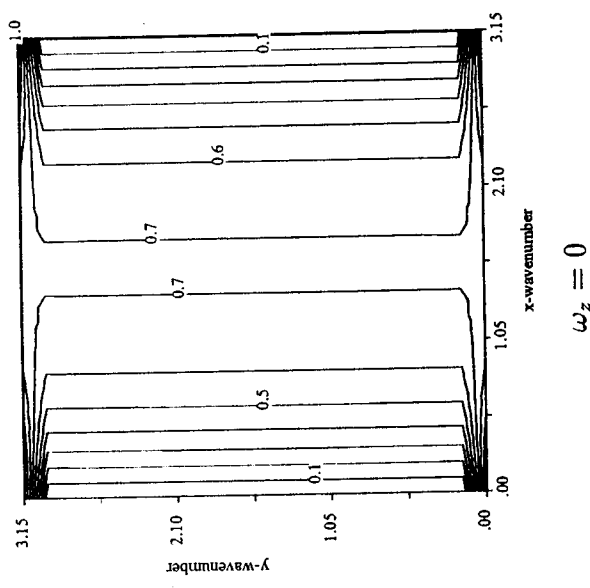
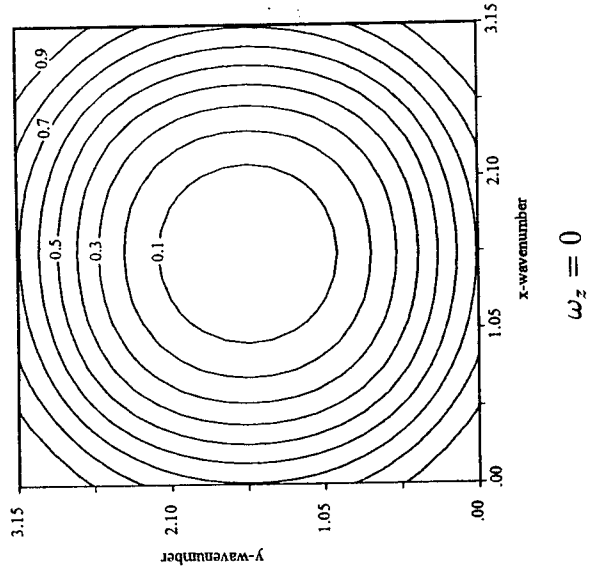


Fig. 9.7. Amplification factor for the dual time step ADI scheme with $\bar{Q}^o = 0$, TYPE I, $\frac{\Delta y}{\Delta z} = 10^4$, $\text{CFL}_z = 1$, $\frac{\Delta t_{yz}}{\Delta t_x} = 1$, $NP = 10$

Fig. 9.8. Amplification factor for the two-factor ADI scheme, TYPE II, $\frac{\Delta x}{\Delta y} = \frac{\Delta x}{\Delta z} = 10^4$, $\text{CFL}_z = 1$

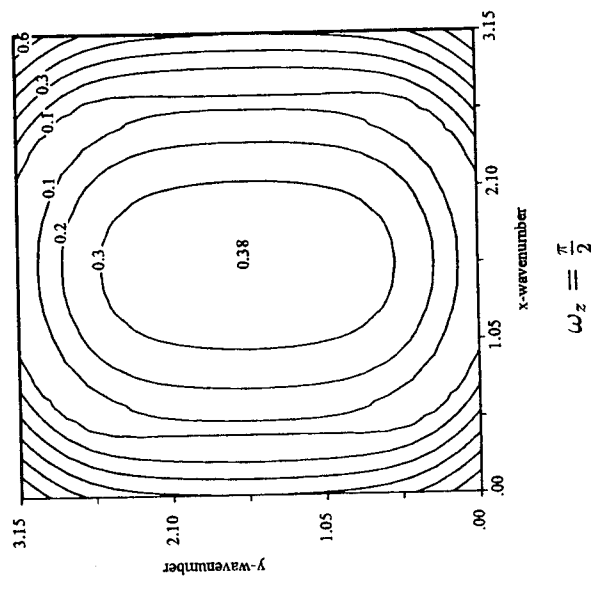
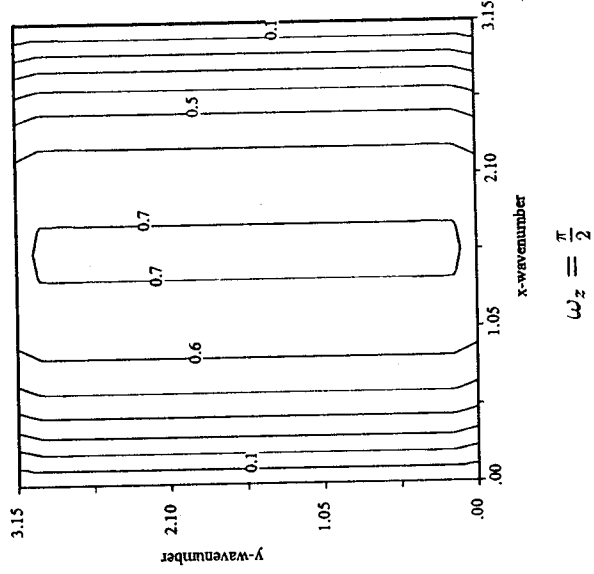
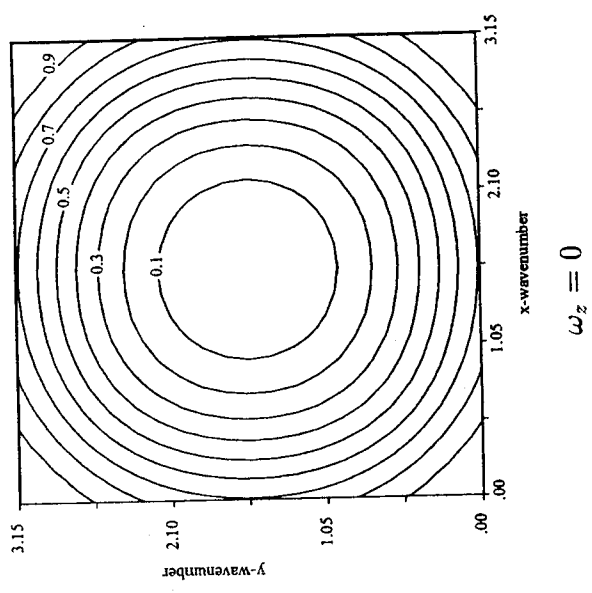
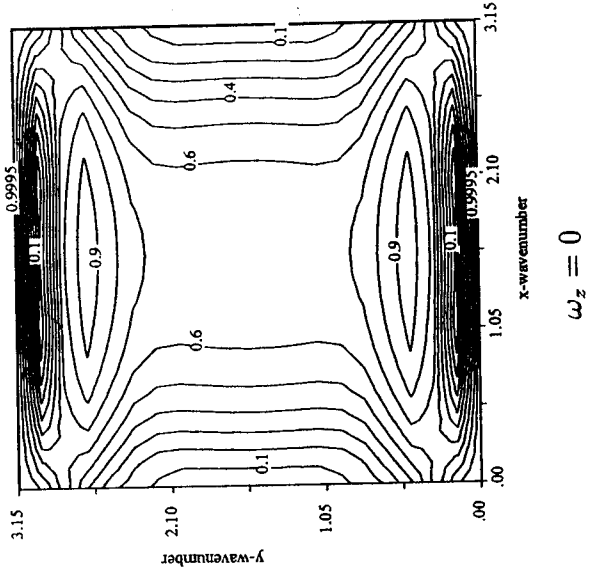


Fig. 9.9. Amplification factor for the dual time step ADI scheme with $\tilde{Q}^o = 0$, TYPE II, $\frac{\Delta x}{\Delta y} = 10^4$, $CFL_z = 1$, $\frac{\Delta t_{yz}}{\Delta t_x} = 10^{-4}$, $NP = 10$

Fig. 9.10. Amplification factor for the dual time step ADI scheme, $\Delta x = \Delta y = \Delta z$, $CFL = 1$, $\frac{\Delta t_{yz}}{\Delta t_x} = 1$, $NP = 1$

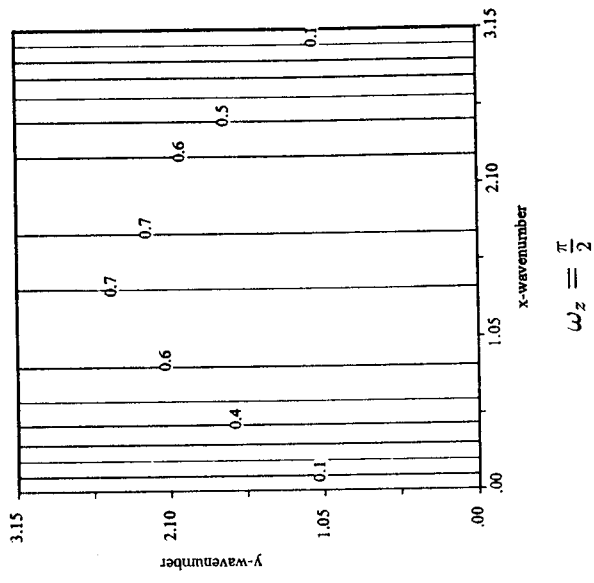
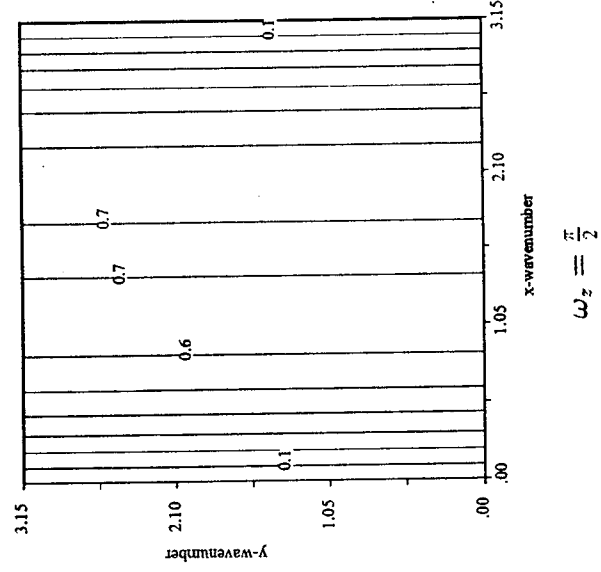
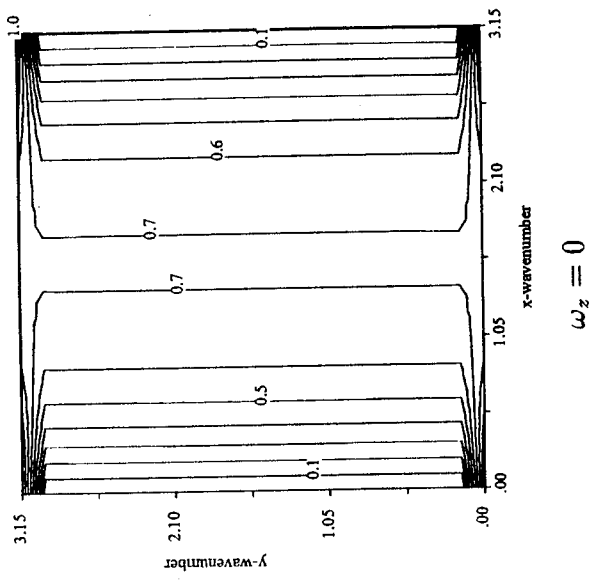
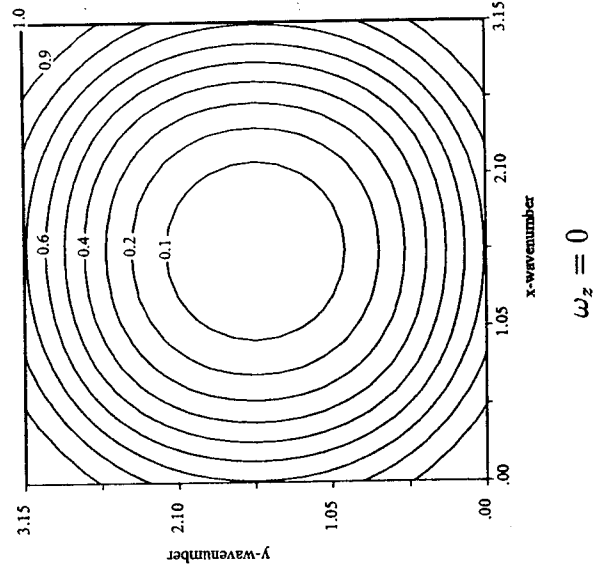


Fig. 9.11. Amplification factor for the dual time step ADI scheme, TYPE I. $\frac{\Delta y}{\Delta z} = \frac{\Delta x}{\Delta z} = 10^4$, $CFL_z = 1$, $\frac{\Delta t_{yz}}{\Delta t_x} = 1$, $NP = 10$

Fig. 9.12. Amplification factor for the dual time step ADI scheme, TYPE II. $\frac{\Delta x}{\Delta y} = \frac{\Delta x}{\Delta z} = 10^4$, $CFL_z = 1$, $\frac{\Delta t_{yz}}{\Delta t_x} = 10^{-4}$, $NP = 10$

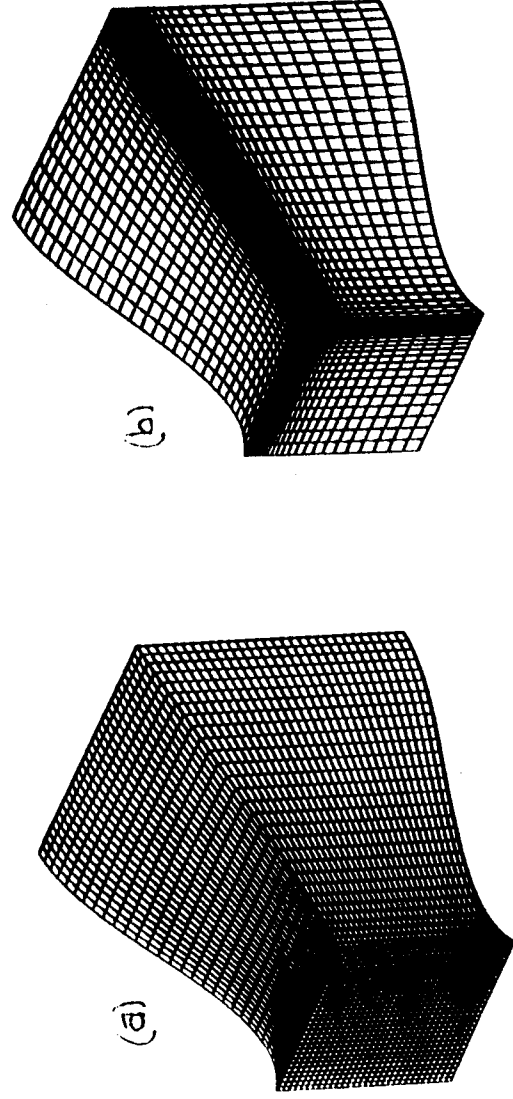


Fig. 9.13. Grid geometry (31x31x31) for flow in three-dimensional choked nozzles: (a) uniform grid, (b) stretched grid.

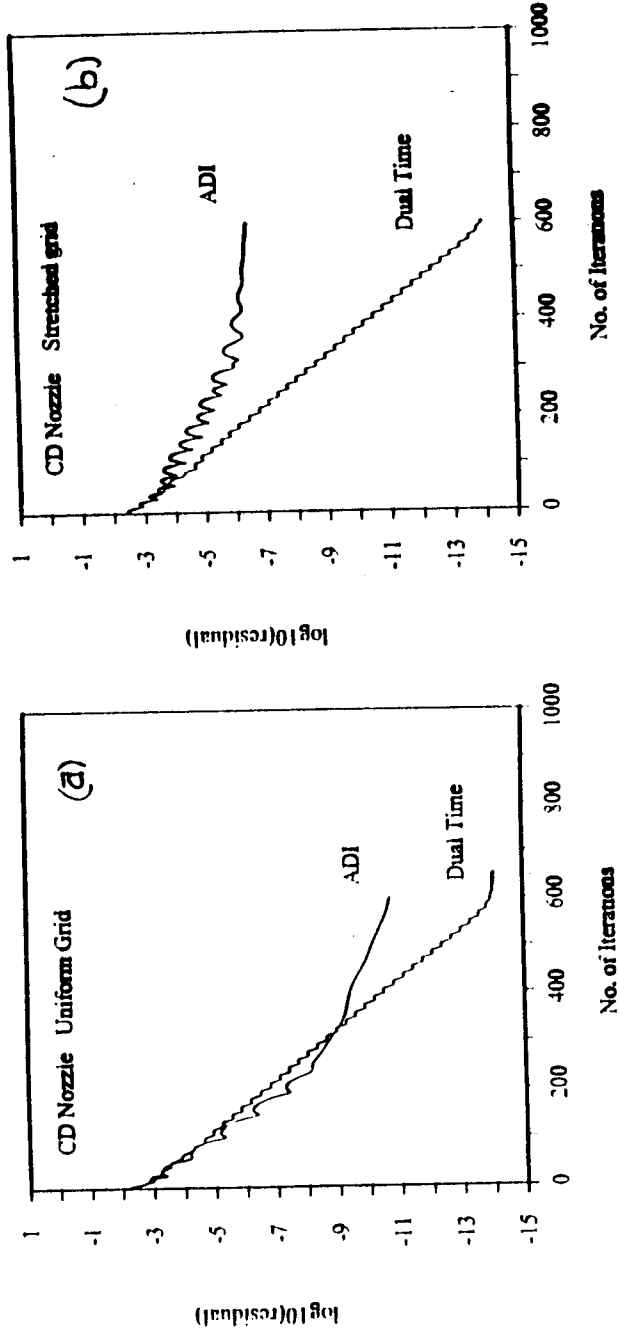


Fig. 9.14. Comparison of convergence of standard three-factor ADI and dual-time schemes for the three-dimensional nozzle: (a) uniform grid, (b) stretched grid.

CONCLUSIONS

The implementation of CFD codes in design applications requires that they be robust and effective over the flow regimes of interest. Most algorithms, however, are effective over rather limited regimes of flow conditions, while design applications frequently involve very broad ranges. This invariably leads to code failures. The present effort was aimed at improving code robustness by: (a) identifying those flow regimes that are encountered in typical rocket propulsion applications; (b) identifying corresponding convergence regimes for representative CFD algorithms; and (c) developing algorithm modifications that enable the algorithms to work effectively over a broader range of parameters, thereby providing convergence enhancement.

Concerning the regimes of application in rocket propulsion problems, it was shown that even for high Reynolds number problems, the grid cells near the walls where strong grid stretching is used to resolve boundary layer become viscous dominated, so that the local convergence is diffusion-dominated rather than convection dominated. At the same time, the larger cells in the center of the flow passages are completely controlled by convective effects so that both "inviscid" convergence and "viscous" convergence must simultaneously be efficient to obtain robustness in high Reynolds number flows. Similarly, many propulsion problems have large regions of low speed flow as well as regions of transonic and supersonic flow, so the computation methods must be uniformly effective over all Mach number regimes as well as over all Reynolds number regimes. In addition, high Reynolds number problems require the use of highly stretched grids, and these high grid aspect ratios can be strongly detrimental to convergence and robustness. Finally, propulsion problems frequently involve finite rate chemical kinetics and almost invariably involve turbulence models. These auxiliary physical effects introduce still more characteristic times and length scales that adversely affect code robustness.

The present effort has been directed toward comparing the capabilities of pressure-based and density-based algorithms with a view toward assessing their effectiveness in solving representative propulsion problems. A second area of focus seeks to expand the capabilities of density-based methods by time-derivative preconditioning so that their domain of effectiveness encompasses the full range of problems encountered in propulsion applications. The application of linear stability theory to the understanding of convergence rates was a central theme of the approach used.

As a comparison of pressure-based and density-based algorithms, complete stability analyses for the full Navier-Stokes equations were developed for both systems. For the pressure-based system, attention was focused on the PISO scheme. The results showed that the PISO scheme was conditionally stable, with better convergence at low speeds than at transonic speeds. While this observation was widely known, the stability analysis also showed that the PISO system was nearly identical to density-based system that used preconditioning.

Preconditioning transformed density-based systems to a form that was similar to pressure-based algorithms at low speeds, thereby giving excellent convergence at low speeds. The advantage of the preconditioned, density-based systems was that they automatically shifted back to non-preconditioned systems at transonic and supersonic speeds, thereby providing uniform stability characteristics (and uniform convergence) at all speeds.

To exploit this capability of preconditioning, methods for extending these preconditioning techniques to wider problem regimes were developed. Viscous preconditioning was enhanced substantially so that preconditioned, density-based systems now are available that provide uniform convergence at all speeds and all Reynolds numbers. Both stability analysis and convergence rates indicate that high aspect ratio grids have a major effect on the convergence of high Reynolds number flowfields. Controlling convergence in such problems requires careful treatment of the boundary conditions and the time step. For both central-differenced and upwind-differenced systems it was shown that the inviscid time step should be based on the smallest CFD, not the largest as is traditionally done. For viscous flows, it was necessary to use a minimum CFD and a maximum von Neumann number to avoid convergence problems arising from viscous-inviscid cross-products in the approximate factorization error.

Stability analysis and convergence checks of upwind systems based on line Gauss-Seidel (LGS), point Gauss-Seidel (LU) and ADI showed that the former two were conditionally stable while the latter (ADI) was unconditionally stable. Tests with numerous problems (stability and convergence) indicated that LGS and ADI are about equally effective (although the LGS method was more reliable) while the LU system appeared to be less robust. As indicated above, the optimum time step for all these upwind schemes was the min-CFL, max-VNN combination.

Preconditioning methods were also developed for unsteady dual-time procedures. It was shown that the dual-time method includes the popular "approximate-Newton" iteration as a special case, and that while the approximate-Newton method is very efficient for flows in which the unsteady acoustic waves are of interest (such as combustion instability calculations), it is not efficient for unsteady flows which are characterized by oscillations at the particle speed (for example, unsteady shear layers or diffusion flames). The preconditioned dual time method developed herein enables effective convergence at both these limits. For acoustic-wave computations, the preconditioning matrix for the pseudo-time term is the identity matrix while for the particle oscillations at low Mach numbers, it is the steady-state preconditioning matrix. An appropriate transition between these limits was also developed as part of the present research and is described herein.

A final topic that was investigated is the effect of grid aspect ratio on the convergence of three-dimensional algorithms. Here, grid aspect ratio effects are described by two independent ratios ($\Delta x/\Delta y$ and $\Delta x/\Delta z$), and these two quantities make grid aspect ratio much more significant in three-dimensions than in two-dimensions. Four high aspect ratio cases can be encountered: Type 0, equal order of magnitude spacing in all three directions; Type I, one short and two long coordinate directions; Type II, two short and one long coordinate intervals; and Type III, one short, one medium, and one long. Stability results show that Type 0 grids are easily controlled by nearly any algorithm. Type I grids have a strong adverse effect on convergence, but can be mitigated by proper preconditioning and time-step control. Types II and III remain a problem to all schemes investigated. Additional work

aimed at understanding these two more complex systems is needed before uniform, reliable convergence can be expected in these types of grids.

Bibliography

- [1] Patankar, S. V., Numerical Heat Transfer and Fluid Flow Series in Computational Methods in Mechanics and Thermal Sciences, McGraw-Hill Book Company, 1980.
- [2] Chen, Y. S. "Compressible and Incompressible Flow Computations with a Pressure Based Method," *AIAA Paper 89-0286*, 1989.
- [3] Briley, W. R., and McDonald, H. "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics vol. 24*, 1977, 372-397.
- [4] Warming, R. F., and Beam, R. M. "On the Construction and Application of Implicit Factored Schemes for Conservation Law," *SIAM-AMS Proceedings vol. 11*, 1978, 85-129.
- [5] Pulliam, T. H., and Steger, J. L. "Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms," *AIAA Paper 85-0360*, 23th *AIAA Aerospace Sciences Meeting*, Reno, NV January 1985.
- [6] Turkel, E. "Review of Preconditioning Methods for Fluid Dynamics," *NASA CR-189712, ICASE Report No. 92-47*, 1992.
- [7] Choi, Y. H., and Merkle, C. L. "The Application of Preconditioning to Viscous Flows," *Journal of Computational Physics vol. 105*, April 1993, 207-223.
- [8] Venkateswaran, S., Weiss, J. M., Merkle, C. L., and Choi, Y. H. "Propulsion-Related Flowfields Using the Preconditioned Navier-Stokes Equations," *AIAA Paper 92-3437*, 1992.
- [9] Shuen, J. S., Chen, K. H., and Choi, Y. H. "A Time-Accurate Algorithm for Chemical Non-Equilibrium Viscous Flows at All Speeds," *AIAA Paper 92-3639*, 1992.
- [10] van Leer, B., Lee, W. T., and Roe, P. L. "Characteristic Time-Stepping or Local Preconditioning of the Euler Equations," *AIAA Paper 91-1552-CP, Computational Fluid Dynamics Conference*. Honolulu, Hawaii 1991.
- [11] Godfrey, A. G., Walters, R. W., and van Leer, B. "Preconditioning for the Navier-Stokes Equations," *AIAA Paper 93-0535*, 1993.
- [12] Merkle, C. L., Venkateswaran, S., and Buelow, P. E. O. "The Relationship Between Pressure-Based and Density-Based Algorithms," *AIAA Paper 92-0425*, 30th *AIAA Aerospace Sciences Meeting*, Reno, NV 1992.

- [13] Douglas, J., and Gunn, J. E. "A General Formulation of Alternating Direction Method — Part I — Parabolic and Hyperbolic Problem," *Numerische Mathematik* vol. 82, 1964, 428–453.
- [14] Harlow, F. H., and Welch, J. E. "Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surfaces," *Physics of Fluids* vol. 8, no. 12, December 1965, 2182–2185.
- [15] Issa, R. I. "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics* vol. 62, 1986, 40–65.
- [16] Issa, R. I., Gosman, A. D., and Watkins, A. P. "The Computation of Compressible and Incompressible Recirculating Flows by a Non-Iterative Implicit Scheme," *Journal of Computational Physics* vol. 62, 1986, 66–82.
- [17] Jiang, Y., Chen, C. P., and Chyu, M. K. "A Numerical Computation of Convective Heat Transfer in Two-Pass Rectangular Channels with a 180-Degree Sharp Turn," *AIAA Paper 90-0355*, 1990.
- [18] Chakravarthy, S. R. "Euler Equations—Implicit Schemes and Boundary Conditions," *AIAA Journal* vol. 21, no. 5, May 1983, 669–706.
- [19] Rai, M. M., and Chaussee, D. S. "New Implicit Boundary Procedures: Theory and Application," *AIAA Journal* vol. 22, no. 8, August 1984, 1094–1100.
- [20] Yu, S. T., Hultgren, L. S., and Liu, N. S. "Direct Calculations of Waves in Fluid Flows Using High-Order Compact Difference Scheme," *AIAA Paper 93-0148*, 1993.
- [21] Baldwin, B. S., and Lomax, H. "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," *AIAA Paper 78-257*, 1978.
- [22] Armaly, B. F., Durst, F., Pereira, J. C. F., and Schonung, B. "Experimental and Theoretical Investigation of Backward-Facing Step Flow," *Journal of Fluid Mechanics* vol. 127, 1983, 473–496.
- [23] Jones, W. P., and Launder, B. E. "The Prediction of Laminarization with a Two-Equation Model of Turbulence," *International Journal of Heat and Mass Transfer* vol. 15, 1972, 301–314.
- [24] Chien, K. "Predictions of Channel and Boundary Layer Flows with a Low-Reynolds Number Turbulence Model," *AIAA Journal* vol. 20, no. 1, 1982, 33–38.
- [25] Withington, J. P., Shuen, J. S., and Yang, V. "A Time Accurate, Implicit Method for Chemically Reacting Flows at All Mach Numbers," *AIAA Paper 91-0581*, *AIAA 11th Aerospace Sciences Meeting*, Reno, NV 1991.
- [26] Molvik, G. A., and Merkle, C. L. "A Set of Strongly Coupled, Upwind Algorithms for Computing Flows in Chemical Nonequilibrium," *AIAA Paper 89-0199*, 1989.

- [27] Merkle, C. L., and Choi, Y. H. "Computation of Low-Speed Compressible Flows with Time-Marching Procedures." *International Journal for Numerical Methods in Engineering* vol. 25, 1985, 293-311.
- [28] Westbrook, C. "Hydrogen Oxidation Kinetics in Gaseous Detonations." *Combustion Science and Technology* vol. 29, no. 1-2, 1982, 67-81.
- [29] Shuen, J. S., and Yoon, S. "Numerical Study of Chemically Reacting Flows Using a Lower-Upper Symmetric Successive Overrelaxation Scheme," *AIAA Journal* vol. 27, no. 12, December 1989, 1752-1760.
- [30] Lai, H. T., and Raju, M. M. "CFD Validation of Subsonic Turbulent Planar Shear Layers," *AIAA Paper 93-1773*, 1993.
- [31] Driver, D. M., and Seegmiller, H. L. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow," *AIAA Journal* vol. 23, no. 2, February 1985.
- [32] Jameson, A., Schmidt, W., and Turkel, E. "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes," *AIAA Paper 81-1259*, 1981.
- [33] Thomas, J. L., van Leer, B., and Walters, R. W. "Implicit Flux-Split Schemes for the Euler Equations," *AIAA Paper 85-1680*, 1985.
- [34] Jameson, A., and Turkel, E. "Implicit Schemes and LU Decompositions," *Mathematics of Computation* vol. 37, no. 156, 1981, 385-397.
- [35] Vanden, K. J., and Whitfield, D. L. "Direct and Iterative Algorithms for the Three-Dimensional Euler Equations," *AIAA Paper 93-3378-CP, Proceedings of the 11th AIAA CFD Conference*, Orlando, FL 1993, pp. 1074-1084.
- [36] Thomas, J. L., and Walters, R. W. "Upwind Relaxation Algorithm for the Navier-Stokes Equations," *AIAA Journal* vol. 25, no. 4, 1987, 527-534.
- [37] Lombard, C. K., Bardina, J., Venkatapathy, E., and Oliger, J. "Multi-Dimensional Formulation of CSCM - An Upwind Flux Difference Eigenvector Split Method for the Compressible Navier-Stokes Equations," *AIAA Paper 83-1895*, 1983.
- [38] Chakravarthy, S. "Relaxation Methods for Unfactored Implicit Upwind Schemes," *AIAA Paper 84-0165*, 1984.
- [39] McCormack, R. W. "Current Status of Numerical Solutions of the Navier-Stokes Equations," *AIAA Paper 85-0032*, 1985.
- [40] Rogers, S. E., Kwak, D., and Kiris, C. "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady-State and Time-Dependent Problems," *AIAA Paper 89-0463*, 1989.

- [41] Buelow, P. E. O., Venkateswaran, S., and Merkle, C. L. "The Effect of Grid Aspect Ratio on Convergence," *AIAA Journal* vol. 32, no. 12, December 1994, 2401-2408.
- [42] Rai, M. M. "Navier-Stokes Simulations of Blade-Vortex Interaction Using High-Order Accurate Upwind Schemes," *AIAA Paper 87-0543*, Reno, NV January 1987.
- [43] Merkle, C. L., and Athavale, M. "A Time Accurate Unsteady Incompressible Algorithm Based on Artificial Compressibility," *AIAA Paper 87-1137*, Honolulu, HA 1987.
- [44] Hosangadi, A., Merkle, C. L., and Turns, S. R. "Analysis of Forced Combusting Jets," *AIAA Journal* vol. 28, no. 8, August 1990.
- [45] Venkateswaran, S., Grenda, J. M., and Merkle, C. L. "Computational Fluid Dynamic Analysis of Liquid Rocket Combustion Instability," *AIAA Paper 91-1609*, June 1991.
- [46] Mathur, S. R., Madavan, N. K., and Rajagopalan, R. G. "Solution-Adaptive Structured-Unstructured Grid Method for Unsteady Turbomachinery Analysis, Part II: Results," *Journal of Propulsion and Power* vol. 10, no. 4, 1994, 585-592.
- [47] Schwer, D. A., Tsuei, H. H., and Merkle, C. L. "Computation and Validation of Spatially Developing Reacting Mixing Layers," *AIAA Paper 95-0261*, 1995.
- [48] Edwards, J. R., and McRae, D. S. "Nonlinear Relaxation Navier-Stokes Solver for for Three-Dimensional, High-Speed Internal Flows," *AIAA Journal* vol. 31, no. 7, July 1993, 1222-1228.

Sensitivity-Based Methods for Convergence of Iterative Algorithms

Final Report

K.-Y. Choi and G. S. Dulikravich

Sensitivity-Based Methods for Convergence Acceleration of Iterative Algorithms

Kwang-Yoon Choi and George S. Dulikravich

Department of Aerospace Engineering

The Pennsylvania State University

University Park, PA 16802, USA

Running title: Sensitivity-Based Acceleration Methods

Key words:

numerical methods

acceleration schemes

fast algorithms

sensitivity analysis

All correspondence should be sent to:

Prof. George S. Dulikravich

Department of Aerospace Engineering

233 Hammond Bldg.

The Pennsylvania State University

University Park, PA 16802, USA

(814) 863-0134 phone (814) 865-7092 FAX gsd@ecl.psu.edu

Abstract

A new method to accelerate the convergence of iterative schemes for the numerical integration of systems of partial differential equations has been developed. The basic idea is that the residual at a grid point depends on the values of the solution vector at the neighboring grid points used in the local discretization approximation. Thus, the new acceleration method is based on the sensitivity of the future residual to the change in the solution vector at the neighboring grid points with the objective to minimize the future residual. The result is a set of optimum iterative relaxation parameters for the entire flow field or for each individual grid line. The method is easy to implement in the existing codes. We have applied it to a finite difference code for two-dimensional incompressible Navier-Stokes equations. Test cases involve laminar and turbulent flows with severe grid clustering and flow separation. The results are compared with those of a basic explicit Runge-Kutta (RK) time-stepping iterative algorithm and with the Implicit Residual Smoothing (IRS) and the Distributed Minimal Residual (DMR) acceleration techniques. The new acceleration scheme is shown to be superior to these methods especially on highly-clustered grids.

1. Introduction

One of the consistent goals in computational fluid dynamics is to improve the efficiency of numerical techniques by reducing the total computing time required by an iterative algorithm. One of the successful, explicit techniques used to integrate Euler and Navier-Stokes equations is the Runge-Kutta (RK) explicit time-stepping algorithm [1]. Several attempts have been made to accelerate the convergence of this method. Some of the more successful acceleration methods are based on local time stepping [1], implicit residual smoothing [1,2], enthalpy damping [1], multi-gridding [2] and preconditioning [3].

Here, we plan to elaborate on an optimum extrapolation technique in order to find optimum relaxation parameters for iterative algorithms. An extrapolation procedure based on the power method and the Minimal Residual Method (MRM) was successfully applied [4] to the multi-grid algorithm. The MRM uses equal optimal weights for the corrections to every equation in a system of partial differential equations that is solved, but it has not been shown to accelerate the scheme without multi-gridding. The Distributed Minimal Residual (DMR) method [5-8] based on the General Nonlinear Minimal Residual (GNLMR) method [9-10] allows each component of the solution vector in a system of equations to have its own sequence of optimized relaxation parameters. The DMR method was found to be capable of reducing the computation time by 10 - 75% depending on the test case and grid used. It is an improvement over a well-known General Minimal Residual (GMRES) method [11] which requires a large amount of computer memory. All these optimum extrapolation methods belong to the general class of Krylov subspace methods.

A new convergence acceleration algorithm based on the sensitivity of future residual to flow variables at the surrounding grid points is presented in this paper [12-16]. The new method was tested on uniform and highly-clustered computational grids and for both low and high Reynolds number flows including flows with separation. It was found to perform consistently better than the IRS and the DMR acceleration methods.

2. Incompressible Navier-Stokes Equations

The two-dimensional, incompressible Navier-Stokes equations with Chorin's artificial compressibility method [17] can be written in conservative vector form as

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \frac{\partial \mathbf{E}}{\partial \xi} + \frac{\partial \mathbf{F}}{\partial \eta} - \mathbf{H} + \mathbf{S} = 0 \quad (1)$$

where τ is the time, Q is the solution vector, E and F are the inviscid flux vectors, H is the vector of viscous terms and S is the vector of source terms. Thus

$$Q = \frac{1}{J} \begin{Bmatrix} \frac{P}{\beta} \\ u \\ v \end{Bmatrix} \quad E = \frac{1}{J} \begin{Bmatrix} U \\ Uu + \xi_x p \\ Uv + \xi_y p \end{Bmatrix} \quad F = \frac{1}{J} \begin{Bmatrix} V \\ Vu + \eta_x p \\ Vv + \eta_y p \end{Bmatrix} \quad (2)$$

Here, U and V are the contravariant velocity components, u and v are the Cartesian velocity components, p is the pressure, ξ and η are the curvilinear, non-orthogonal, boundary conforming grid coordinates, J is the determinant of the Jacobian transformation matrix, $J = \det(\partial(\xi, \eta)/\partial(x, y))$, and β is the user-specified artificial compressibility parameter [17]. The term $\partial(\frac{P}{J\beta})/\partial\tau$ was added to the mass conservation to make the entire system non-singular. Adding the non-physical term, $\partial(\frac{P}{J\beta})/\partial\tau$, does not affect the steady-state solution since all the time derivative terms vanish as the solution converges to a steady-state.

3. Implicit Residual Smoothing (IRS) Method [2]

Residual smoothing permits the use of higher values of Courant-Friedrichs-Lewy (CFL) number thus accelerating the convergence. The implicit form of the residual smoothing for a two-dimensional case can be written as

$$[1 - \epsilon\delta_\xi^2][1 - \epsilon\delta_\eta^2]\bar{R} = R \quad (3)$$

Here, R is the original unsmoothed residual, $\bar{R}$ is the smoothed residual, δ^2 is the second order differential operator and ϵ is a user specified coefficient so that

$$\epsilon \geq \frac{1}{4} \left[\left(\frac{CFL}{CFL^*} \right)^2 - 1 \right] \quad (4)$$

Here, CFL^* is the Courant-Friedrichs-Lewy number for the unsmoothed scheme and CFL is the modified Courant-Friedrichs-Lewy number. The IRS is applied at each stage of the RK scheme. It was found that the computing time was increased by a factor of three if the IRS is applied at every iteration together with the current four-stage RK time-stepping. For the test cases investigated in the paper, the IRS was applied after every 10 iterations for optimum performance.

4. Distributed Minimal Residual (DMR) Method [5-8]

This method extrapolates the solution at iteration level $t+1$ from the previous N iteration levels. Specifically, the DMR is presently formulated as

$$\begin{aligned} Q_1^{p+1} &= Q_1^p + \omega_1^p \delta Q_1^p + \omega_1^{p-1} \delta Q_1^{p-1} + \dots + \omega_1^{p-(N-1)} \delta Q_1^{p-(N-1)} \\ &\vdots \\ Q_M^{p+1} &= Q_M^p + \omega_M^p \delta Q_M^p + \omega_M^{p-1} \delta Q_M^{p-1} + \dots + \omega_M^{p-(N-1)} \delta Q_M^{p-(N-1)} \end{aligned} \quad (5)$$

Here, ω 's are the iterative relaxation parameters (weight factors) to be calculated and optimized, δQ 's are the corrections computed with the non-accelerated iteration scheme, N denotes the total number of consecutive iteration steps combined when evaluating the optimum ω 's and M stands for the total number of equations in the system that is being iteratively solved. The DMR method calculates optimum ω 's to minimize the L-2 norm of the future residual of the system integrated over the entire

domain. The present formulation of the DMR uses the same values of the $N \times M$ optimized relaxation parameters at every grid point, although different parts of the flow field converge at different rates.

5. Sensitivity-Based Minimal Residual (SBMR) Method [12-15]

Conventional iterative algorithms update the flow variables by calculating the amount of corrections without optimizing their influence on the future residual. The SBMR method evaluates the sensitivity of the residual to the solution vectors at surrounding grid points and calculates the optimum amount of corrections necessary to minimize the overall future residual. The basic idea is that the residual at a grid point depends on the values of the solution vector Q at the neighboring grid points used in the local finite difference approximation. The rate at which the residual, R_m ($m = 1, \dots, M$; number of equations in the system to be iteratively solved), changes with Q_s ($s = 1, \dots, S$;

number of neighboring grid points involved in the local discretization scheme) is $\frac{\partial R_m}{\partial Q_s}$. These

sensitivities can be determined by taking partial derivatives of the finite difference approximation of the residual with respect to the solution vector Q .

Suppose we know the flow variables Q at iteration level t and at $t+n$ where n is the number of regular iteration steps performed by the original non-accelerated code. Then, the changes in the components of Q between the two iteration levels in case of the two-dimensional, incompressible Navier-Stokes equations involving mass, x-momentum and y-momentum conservation equations (2) are given as

$$\Delta p_s = (p_s)^{t+n} - (p_s)^t \quad \Delta u_s = (u_s)^{t+n} - (u_s)^t, \quad \Delta v_s = (v_s)^{t+n} - (v_s)^t, \quad (6)$$

Using the first two terms of a Taylor series expansion in the artificial time direction, the residual for each of the equations in the system given by equation (2) after n iterations will be

$$R_m^{t+n} = R_m^t + \left[\sum_s \frac{\partial R_m^t}{\partial p_s} \Delta p_s \right] + \left[\sum_s \frac{\partial R_m^t}{\partial u_s} \Delta u_s \right] + \left[\sum_s \frac{\partial R_m^t}{\partial v_s} \Delta v_s \right] \quad (7)$$

If we introduce weighting factors α_p , α_u and α_v to corrections Δp , Δu and Δv respectively, the future solution vector components can be estimated as

$$\begin{aligned} (p_j)^{t+n+1} &= (p_j)^t + \alpha_p \Delta p_s \\ (u_j)^{t+n+1} &= (u_j)^t + \alpha_u \Delta u_s \\ (v_j)^{t+n+1} &= (v_j)^t + \alpha_v \Delta v_s \end{aligned} \quad (8)$$

Subsequently, the future residual at $(t+n)+1$ can be approximated by

$$R_m^{(t+n)+1} = R_m^t + \left[\sum_s \frac{\partial R_m^t}{\partial p_s} (\Delta p_s \alpha_p) \right] + \left[\sum_s \frac{\partial R_m^t}{\partial u_s} (\Delta u_s \alpha_u) \right] + \left[\sum_s \frac{\partial R_m^t}{\partial v_s} (\Delta v_s \alpha_v) \right] \quad (9)$$

For now, each of the α 's is assumed to have the same value over the whole domain, D. Therefore, equation (9) can be written as

$$R_m^{(t+n)+1} = R_m^t + \left[\sum_s \frac{\partial R_m^t}{\partial p_s} \Delta p_s \right] \alpha_p + \left[\sum_s \frac{\partial R_m^t}{\partial u_s} \Delta u_s \right] \alpha_u + \left[\sum_s \frac{\partial R_m^t}{\partial v_s} \Delta v_s \right] \alpha_v \quad (10)$$

Subtracting (7) from (10) we have

$$R_m^{(t+\eta+1)} = R_m^{t+n} + \left[\sum_s \frac{\partial R_m^t}{\partial p_s} \Delta p_s (\alpha_p - 1) + \left[\sum_s \frac{\partial R_m^t}{\partial u_s} \Delta u_s (\alpha_u - 1) + \left[\sum_s \frac{\partial R_m^t}{\partial v_s} \Delta v_s (\alpha_v - 1) \right] \right] \right] \quad (11)$$

The α 's are determined such that the L-2 norm of the future residual over the entire domain D will be minimized. In general, this means that

$$\sum_D \left[\frac{\partial \left(\sum_{m=1}^M (R_m^{(t+\eta+1)})^2 \right)}{\partial \alpha_q} \right] = 2 \sum_D \left[\sum_{m=1}^M R_m^{(t+\eta+1)} \frac{\partial R_m^{(t+\eta+1)}}{\partial \alpha_q} \right] = 0 \quad (12)$$

where q stands for each primitive flow variable. For simplicity, let us denote the known bracketed terms in equation (11) as a_{mp} , a_{mu} and a_{mv} so that

$$R_m^{(t+\eta+1)} = R_m^{t+n} + a_{mp}(\alpha_p - 1) + a_{mu}(\alpha_u - 1) + a_{mv}(\alpha_v - 1) \quad (13)$$

Substituting (13) into (12) gives the following equations for optimal global α 's pertinent to the system given by equation (1).

$$\begin{aligned} \sum_D \left[\sum_{m=1}^3 \{ R_m^{t+n} + a_{mp}(\alpha_p - 1) + a_{mu}(\alpha_u - 1) + a_{mv}(\alpha_v - 1) \} a_{mp} \right] &= 0 \\ \sum_D \left[\sum_{m=1}^3 \{ R_m^{t+n} + a_{mp}(\alpha_p - 1) + a_{mu}(\alpha_u - 1) + a_{mv}(\alpha_v - 1) \} a_{mu} \right] &= 0 \\ \sum_D \left[\sum_{m=1}^3 \{ R_m^{t+n} + a_{mp}(\alpha_p - 1) + a_{mu}(\alpha_u - 1) + a_{mv}(\alpha_v - 1) \} a_{mv} \right] &= 0 \end{aligned} \quad (14)$$

In equation (14), R's and a's are known from the past iteration results. Since each α is assumed to have the same value over the entire computational domain, equation (14) gives a system of three simultaneous algebraic equations for α_p , α_u and α_v .

$$\begin{aligned} \left[\sum_D \left(\sum_{m=1}^3 a_{mp} a_{mp} \right) \right] (\alpha_p - 1) + \left[\sum_D \left(\sum_{m=1}^3 a_{mu} a_{mu} \right) \right] (\alpha_u - 1) + \left[\sum_D \left(\sum_{m=1}^3 a_{mv} a_{mv} \right) \right] (\alpha_v - 1) &= - \sum_D \left(\sum_{m=1}^3 R_m^u a_{mp} \right) \\ \left[\sum_D \left(\sum_{m=1}^3 a_{mp} a_{mu} \right) \right] (\alpha_p - 1) + \left[\sum_D \left(\sum_{m=1}^3 a_{mu} a_{mu} \right) \right] (\alpha_u - 1) + \left[\sum_D \left(\sum_{m=1}^3 a_{mv} a_{mv} \right) \right] (\alpha_v - 1) &= - \sum_D \left(\sum_{m=1}^3 R_m^u a_{mu} \right) \\ \left[\sum_D \left(\sum_{m=1}^3 a_{mp} a_{mv} \right) \right] (\alpha_p - 1) + \left[\sum_D \left(\sum_{m=1}^3 a_{mu} a_{mv} \right) \right] (\alpha_u - 1) + \left[\sum_D \left(\sum_{m=1}^3 a_{mv} a_{mv} \right) \right] (\alpha_v - 1) &= - \sum_D \left(\sum_{m=1}^3 R_m^u a_{mv} \right) \end{aligned} \quad (15)$$

In the general case of a system (1) composed of M partial differential equations, there will be M constant optimum values of α . This means that the system (15) will become a full MxM matrix for M unknown optimum α 's. Also, each summation in the system (15) which in this example is from $m = 1$ to 3, will become a summation from $m = 1$ to M.

Each application of the SBMR creates significant disturbances in the eigenvalues of the iterative matrix. Thus, the optimum values of α 's are determined and applied only periodically followed by a number of iterations with the basic non-accelerated algorithm. This serves as a form of smoothing since too frequent application of the SBMR can lead to instabilities and divergence.

6. Line SBMR (LSBMR) Method [15,16]

The SBMR method calculates the optimum weighting factors α for corrections to the flow variables during the iteration procedure. Since the previous formulation assumes these factors to be constant over the whole computational domain, it should be classified as global SBMR method [12,

14]. For non-uniform grids and rapidly-varying flow variables, optimum α 's should not necessarily be the same over the whole computational domain. A modified formulation (line SBMR or LSBMR) will be elaborated upon to allow these α 's to have different values [15, 16] from one grid line to another. The formulation will be explained using the two-dimensional, incompressible Navier-Stokes equations.

Let the grid be clustered in the j -direction and let each $j = \text{constant}$ grid line have its own set of constant α 's. The residual at a point (i,j) incorporates α 's at the neighboring grid points plus the point (i,j) . For the system given by equation (1) this results in

$$\begin{aligned}
 R_m^{(i+j+1)} = & R_m^i + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_m^i}{\partial p_{s,j-1}} \Delta p_{s,j-1} \alpha_{s,j-1}^{j-1} + \frac{\partial R_m^i}{\partial p_{s,j}} \Delta p_{s,j} \alpha_{s,j}^j + \frac{\partial R_m^i}{\partial p_{s,j+1}} \Delta p_{s,j+1} \alpha_{s,j+1}^{j+1} \right] \\
 & + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_m^i}{\partial u_{s,j-1}} \Delta u_{s,j-1} \alpha_{s,j-1}^{j-1} + \frac{\partial R_m^i}{\partial u_{s,j}} \Delta u_{s,j} \alpha_{s,j}^j + \frac{\partial R_m^i}{\partial u_{s,j+1}} \Delta u_{s,j+1} \alpha_{s,j+1}^{j+1} \right] \\
 & + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_m^i}{\partial v_{s,j-1}} \Delta v_{s,j-1} \alpha_{s,j-1}^{j-1} + \frac{\partial R_m^i}{\partial v_{s,j}} \Delta v_{s,j} \alpha_{s,j}^j + \frac{\partial R_m^i}{\partial v_{s,j+1}} \Delta v_{s,j+1} \alpha_{s,j+1}^{j+1} \right]
 \end{aligned} \tag{16}$$

On each $j = \text{constant}$ grid line, three values of constant α 's are determined in such a way as to minimize the L-2 norm of the future global residual.

$$\begin{aligned}
 2 \sum_D \left(R_1^{i+1} \frac{\partial R_1^{i+1}}{\partial \alpha_p^j} + R_2^{i+1} \frac{\partial R_2^{i+1}}{\partial \alpha_p^j} + R_3^{i+1} \frac{\partial R_3^{i+1}}{\partial \alpha_p^j} \right) &= 0 \\
 2 \sum_D \left(R_1^{i+1} \frac{\partial R_1^{i+1}}{\partial \alpha_u^j} + R_2^{i+1} \frac{\partial R_2^{i+1}}{\partial \alpha_u^j} + R_3^{i+1} \frac{\partial R_3^{i+1}}{\partial \alpha_u^j} \right) &= 0
 \end{aligned}$$

$$2 \sum_D \left(R_1^{j+1} \frac{\partial R_1^{j+1}}{\partial \alpha_v^j} + R_2^{j+1} \frac{\partial R_2^{j+1}}{\partial \alpha_v^j} + R_3^{j+1} \frac{\partial R_3^{j+1}}{\partial \alpha_v^j} \right) = 0 \quad (17)$$

If jmax is the total number of $j = \text{constant}$ grid lines, then substituting the equation (16) into (17) results in $(j_{\text{max}} - j_{\text{min}} + 1) \times 3$ algebraic equations for the same number of unknown α 's. For a given $j = \text{constant}$ grid line the three values of α^j appear only in $R_m^{j+1}(i, j-1)$, $R_m^{j+1}(i, j)$ and $R_m^{j+1}(i, j+1)$. The summation over the entire domain in equation (17) leaves the terms only with α^{j-2} , α^{j-1} , α^j , α^{j+1} and α^{j+2} . At the solid walls ($j = j_{\text{min}}$ and $j = j_{\text{max}}$) the velocity components are zero and no corrections are needed resulting in $\alpha_u^{j_{\text{min}}} = \alpha_v^{j_{\text{min}}} = \alpha_p^{j_{\text{min}}} = \alpha_u^{j_{\text{max}}} = \alpha_v^{j_{\text{max}}} = \alpha_p^{j_{\text{max}}} = 0$. For the sake of computational simplicity we used $\alpha_p^{j_{\text{min}}} = \alpha_p^{j_{\text{min}}+1}$ and $\alpha_p^{j_{\text{max}}} = \alpha_p^{j_{\text{max}}-1}$. In this example the simultaneous system of equations (17) yields a block penta-diagonal matrix equation for $(j_{\text{max}} - j_{\text{min}} + 1) \times 3$ optimum α 's where each block is a 3×3 matrix. In the general case of a two-dimensional system (1) having M partial differential equations, the block penta-diagonal system (17) will have blocks of $M \times M$ size. In the case of a three-dimensional system (1), the system (17) becomes a block septa-diagonal matrix.

7. Computational Results

Computational results are given for a steady, incompressible, viscous flow in a two-dimensional straight channel and a U-shaped turn-around channel. Convergence histories of the basic four-stage RK scheme with local time-stepping, and four acceleration methods (IRS, DMR, SBMR and LSBMR) were compared for two different Reynolds numbers with several grid clusterings. For the laminar flow examples, a fully developed parabolic axial velocity (u) profile and zero lateral velocity (v) were specified at the inlet of the channel. In the rest of the flow domain the initial guess for the

pressure, v-velocity and u-velocity components were $p = 0$, $v = 0$ and $u = 1.0 \times 10^{-5}$, respectively. The inlet pressure was iteratively computed by enforcing characteristic boundary conditions at the inlet for both laminar and turbulent flows. At the exit boundary we used non-reflecting [18, 19] boundary conditions since there was a recirculating flow at the exit of the U-shape channel. Both velocity components were set to zero at the channel walls. Wall pressures were computed from the normal momentum equation. No artificial viscosity was used in any of the test cases.

One of the basic objectives of the present study is to develop a method that is capable of accelerating iterative convergence rates on clustered computational grids. For a straight channel, uniform grid spacing was used in the flow direction clustering the grid symmetrically towards lower and upper solid walls. Maximum grid cell aspect ratio ($\Delta x / \Delta y$) occurs on the upper and lower solid boundaries. For different degrees of clustering, the maximum cell aspect ratio on the solid wall was specified and then the grid spacing was increased towards the center of the channel using a geometric progression

$$\Delta y_{j+1} = c \Delta y_j \quad (18)$$

The factor c was calculated to satisfy the given maximum cell aspect ratio, $AR_{\max}$, and the channel height, H . For a U-shaped channel, a clustering function [20] of the following type was used

$$y = H \frac{(\kappa + 2\delta) \left[\frac{\kappa + 1}{\kappa - 1} \right]^{\frac{\bar{y} - \delta}{1 - \delta}} - \kappa + 2\delta}{(2\delta + 1) \left\{ 1 + \left[\frac{\kappa + 1}{\kappa - 1} \right]^{\frac{\bar{y} - \delta}{1 - \delta}} \right\}} \quad (19)$$

and
$$\bar{x} = \frac{L}{j_{\max} - j_{\min}} \quad \bar{y} = \frac{H}{j_{\max} - j_{\min}} \quad (20)$$

Here, $\delta = 0.5$ was used to have grid lines symmetrically clustered towards the upper and lower walls. For 129 x 80 grid cells, $\kappa=1.066$ was used to obtain the maximum grid aspect ratio $AR_{\max}$ = 40, while $\kappa=1.00774$ was used for obtaining $AR_{\max} = 200$.

The first example is for a low Reynolds number ($Re = 1600$) laminar flow with a mild grid clustering (maximum cell aspect ratio $AR_{\max} = 10$). For the IRS in this case it was found that $\epsilon = 2.5$ produced the fastest convergence. The DMR and the SBMR methods were applied once after every 30 iterations, while the LSBMR method was applied once after every 100 iterations. The DMR method combined three consecutive previous iteration results, while the SBMR and LSBMR methods utilized results from two solutions at 10 and 20 iteration steps apart, respectively. The IRS reduced the number of iterations by 44%, while the DMR, LSBMR and SBMR methods reduced the number of iterations by 53%, 65% and 79%, respectively (Fig. 1). For this mild grid clustering, the LSBMR method was not as efficient as the SBMR method, but better than the existing IRS and DMR methods. Computing time overheads for the SBMR and the LSBMR methods were negligible, while those of the IRS and the DMR methods were not. In terms of computing time reduction, 34%, 49%, 65% and 78% of savings were achieved by the IRS, DMR, LSBMR and SBMR methods respectively. Thus, convergence histories versus computational time consistently mirror the convergence histories versus number of iterations.

As the maximum cell aspect ratio increased to $AR_{\max} = 100$, the LSBMR method converged faster than the SBMR method (Fig. 2). This was an expected result, since the LSBMR method allows different optimum α 's in the clustered region rather than enforcing fixed acceleration parameters over the whole computational domain. To access the accuracy of our algorithm, the computational results were compared with analytical solutions. Figure 3a shows the relative error in pressure drop at different axial locations. Considering the constant axial pressure gradient, dp/dx , was of the order of 10^{-3} for this Reynolds number, we found that the absolute error in the pressure drop was close to machine zero. In Figure 3b, axial velocity error is of the order of 10^{-12} and shows symmetrical distribution about the channel center line which demonstrates that the mass flow

is conserved at each cross section. The analytical value of the v -velocity should be zero everywhere and the computed values were of the order of 10^{-13} .

Further increase in maximum cell aspect ratio ($AR_{\max} = 10,000$) slows down the overall convergence. However, the LSBMR method maintains a faster convergence, while other schemes fail to accelerate the basic iteration method (Fig. 4). The history of acceleration coefficients of the LSBMR method demonstrate that the acceleration coefficients in the fine grid region are much larger (Fig. 5a) than those in the coarse grid region (5b). It should also be noticed that all the acceleration coefficients approach the same value as the solution converges to the machine accuracy.

The convergence acceleration schemes were then tested for a high Reynolds number ($Re = 1.6$ million) turbulent flow in a straight channel. For this high Reynolds number case, we used the Baldwin-Lomax [21] turbulence model and non-reflecting boundary condition was used at the exit of the channel to predict pressure at the exit plane. The inlet flow was assumed to be fully developed, although unlike for the laminar flow cases, the turbulent inlet velocity profile cannot be given analytically. The inlet axial velocity profile was initially assumed to be of the $1/7$ power of y/δ and the initial guess for pressure and velocity components throughout the domain were the same as for the laminar flow examples. We often predicted pressure peaks at the inlet corners when the inlet velocity profile was specified and kept fixed. These pressure peaks were found to be sensitive to the specified inlet velocity profile and their magnitudes increased as the grid became more clustered. Unfortunately, these pressure peaks significantly slowed down the convergence. To circumvent this problem, the inlet velocity profile was slightly modified after each iteration by replacing it with an average of the computed velocity profiles at several immediate locations downstream from the inlet. With this minor modification, the pressure peaks at the inlet corners were quickly eliminated and convergence and robustness of the code significantly improved. With this grid clustering, several grid points could be located within the laminar sublayer. The LSBMR method used the results at 30 iteration levels apart and $\epsilon = 0.5$ was used for the IRS. Convergence histories for $AR_{\max} = 1000$ demonstrate (Fig. 6) that the LSBMR method consistently

outperformed other acceleration schemes. When the maximum cell aspect ratio was increased to $AR_{max} = 10,000$, the LSBMR converged consistently faster than other schemes (Fig. 7).

Fig. 6
Fig. 7

As an example of a complex flow with a separated region inside the computational domain, a flow in a U-shaped channel was calculated. The geometry of the channel was discretized with 129 x 80 grid cells that were symmetrically clustered towards the channel walls. In this test case we used $Re = 200$, $CFL = 2.8$, von Neumann number $= 0.4$ and $\beta = 10$. Anticipating a recirculating flow pattern at the exit of the U-shaped channel, we used a non-reflecting [14, 15] boundary condition at the exit boundary. A high-pressure region was found at the outer wall of the turning

Fig. 8

section while a low-pressure region was located at the inner part (Fig. 8a). The adverse pressure gradient along the inner part of the turning section resulted in an open-end flow separation downstream of the turning section (Fig. 8b) which was successfully predicted because we used the correct form of the nonreflecting exit boundary condition. The LSBMR method used the results at 30 iteration levels apart. The convergence histories for the maximum cell aspect ratio of $AR_{max} = 40$ for this complex flow example show (Fig. 9) that IRS (with $\epsilon = 0.1$ for the best performance with $AR_{max} = 40$) performed only slightly better than the basic RK method. The LSBMR and the

Fig. 9
Fig. 10

DMR methods reduced the number of iterations by 34%, while the DMR reduced it by 20%.

When the grid was further clustered toward the channel walls to give maximum cell aspect ratio of $AR_{max} = 200$, LSBMR method maintained the fastest convergence (Fig. 10), while the DMR and

the SBMR methods were not as fast as in the previous lower cell aspect ratio case. Again, the IRS (with $\epsilon = 1.0$ for the best performance with $AR_{max} = 200$) did not yield any noticeable acceleration

for this test case.

8. Conclusions

The Sensitivity-Based Minimum Residual (SBMR) and the Line SBMR (LSBMR) methods were developed and applied to accelerate the convergence of the explicit RK algorithm for

incompressible Navier-Stokes equations. The methods are easy to comprehend and to implement in the existing computer codes for iterative integration of systems of partial differential equations. Both new acceleration methods consistently enhanced the convergence rate of the basic Runge-Kutta (RK) method and outperformed the Distributed Minimal Residual (DMR) and the Implicit Residual Smoothing (IRS) methods for both laminar and turbulent flows including separation. The advantage of using the LSBMR method became more evident with increased grid clustering. Both new methods require less computer memory than the DMR method. It can be concluded that the SBMR and the LSBMR methods enhance efficiency and robustness of the CFD codes even on highly-clustered grids.

Acknowledgments

This work was supported by the grant NAS 8-38861 from NASA MSFC, Huntsville, AL monitored by Dr. Paul McConnaughey and by the Penn State NASA Propulsion Engineering Research Center, contract NAGW-1356, Supplement 8. Computing time was provided on the NASA Ames NAS Cray-90 facility by Dr. Louis Povinelli of NASA Lewis Research Center. The final text proofreading was performed by Mr. Thomas J. Martin and Ms. Amy Myers.

References

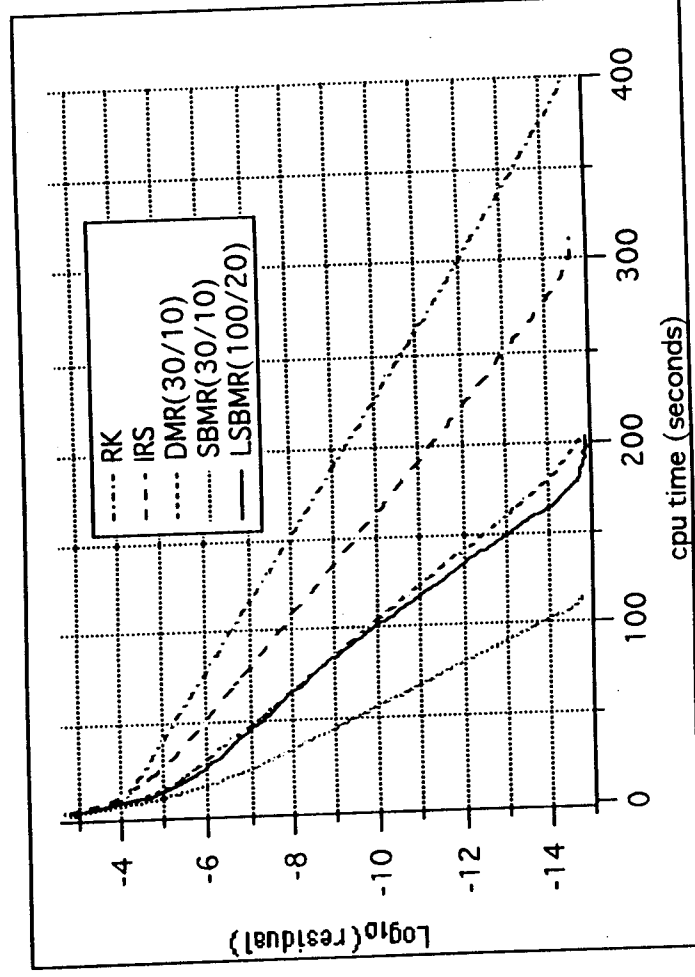
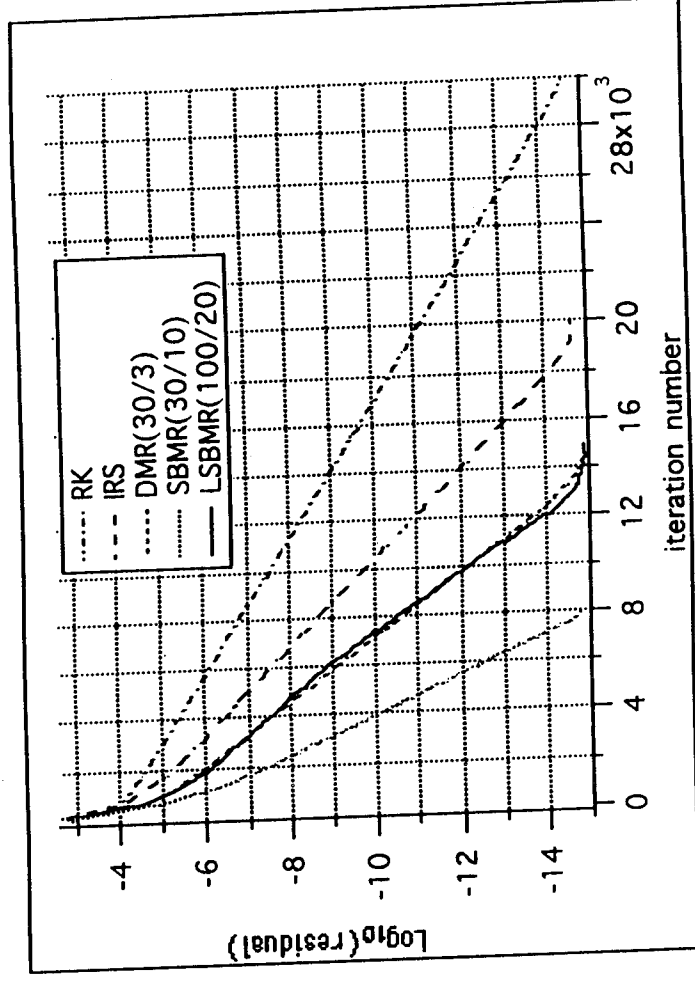
- [1] A. Jameson, W. Schmidt and E. Turkel, Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, AIAA Paper 81-1259, Palo Alto, CA (June 1981).

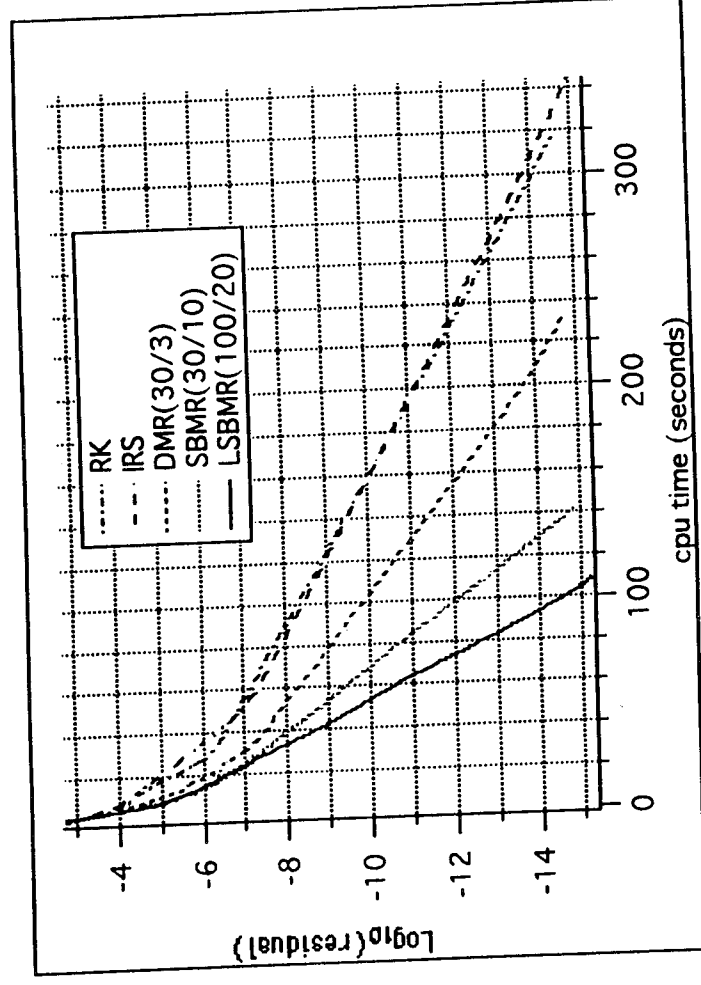
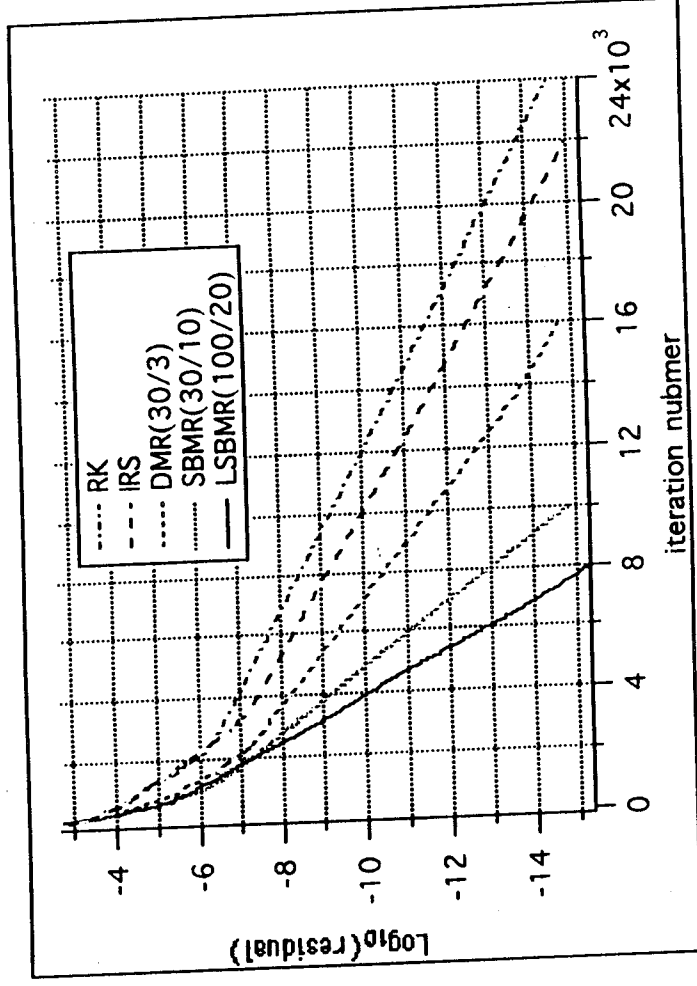
- [2] A. Jameson and T.J. Baker, Solution of the Euler Equations for Complex Configurations, Proceedings of the AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, (June 1983) 293-302.
- [3] E. Turkel, Review of Preconditioning Methods for Fluid Dynamics, ICASE Report No. 92-47, NASA Langley Research Center, Hampton, VA (September 1992).
- [4] M. Hafez, E. Parlette and M.D. Salas, Convergence Acceleration of Iterative Solutions of Euler Equations for Transonic Flow Computations, AIAA Paper 85-1641, Cincinnati, OH (July 1985).
- [5] S. Lee and G.S. Dulikravich, Acceleration of Iterative Algorithms for Euler Equations of Gasdynamics, AIAA Journal, 28 (5) (May 1990) 939-942.
- [6] S. Lee and G.S. Dulikravich, Distributed Minimal Residual (DMR) Method for Acceleration of Iterative Algorithms, Computer Methods in Applied Mechanics and Engineering, 86 (1991) 245-262.
- [7] S. Lee and G.S. Dulikravich, Accelerated Computation of Viscous, Steady Incompressible Flow, ASME Paper 89-GT-45, Gas Turbine and Aeroengine Congress and Exposition, Toronto, Canada (June 4-8, 1989).
- [8] S. Lee and G.S. Dulikravich, Accelerated Computation of Viscous Flow with Heat Transfer, Numerical Heat Transfer: Fundamentals, Part B, 19 (June 1991) 223-241.
- [9] C.-Y. Huang and G.S. Dulikravich, Fast Iterative Algorithms Based on Optimized Explicit Time-Stepping, Computer Methods in Applied Mechanics and Engineering, 63 (August 1987) 15-36.
- [10] S.R. Kennon and G.S. Dulikravich, Optimum Acceleration Factors for Iterative Solution of Linear and Non-Linear Differential Systems, Computer Methods in Applied Mechanics and Engineering, 47 (1984) 357-367.
- [11] Y. Saad and M. Schultz, Conjugate Gradient-Like Algorithms for Solving Non-symmetric Linear Systems, Mathematics of Computation, 44, (170) (1985) 417-424.

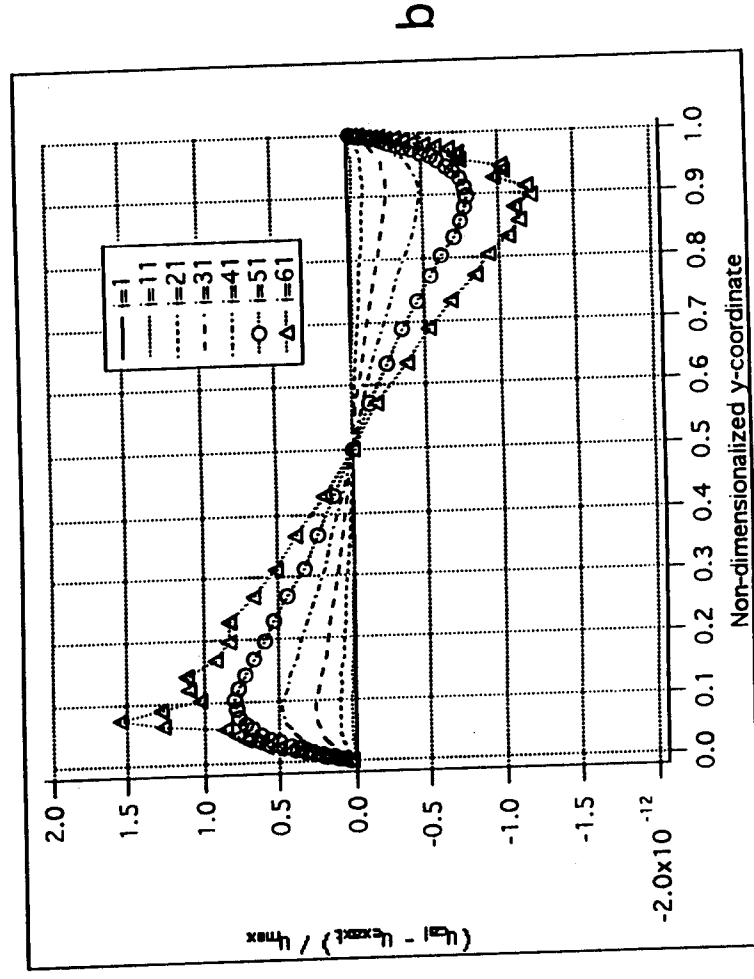
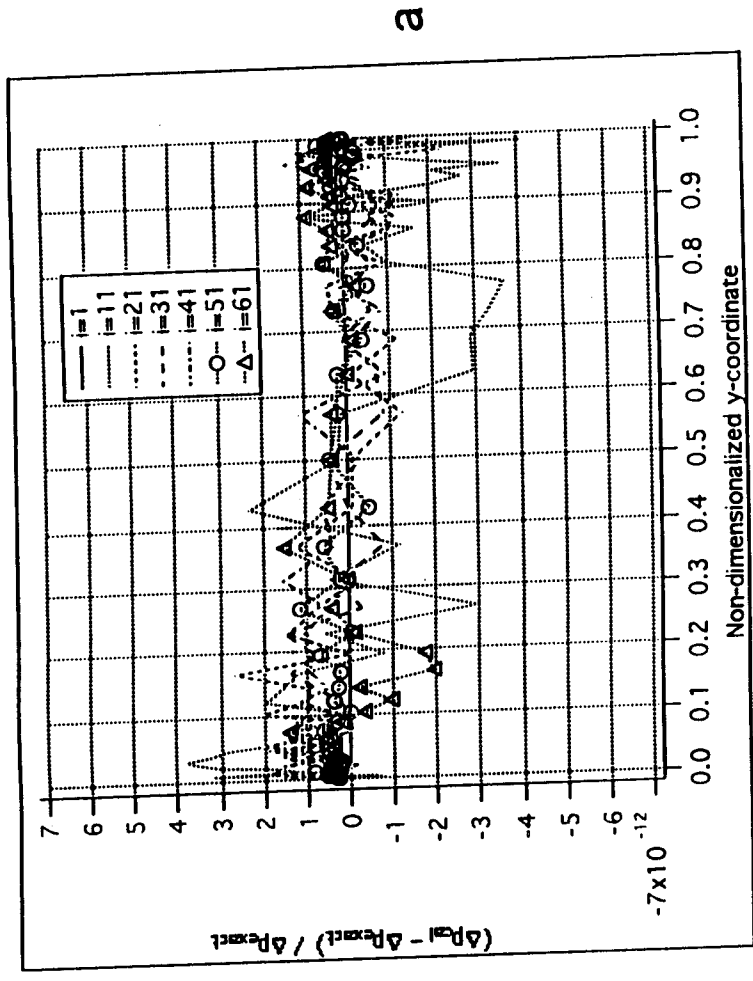
- [12] K.-Y. Choi and G.S. Dulikravich, Reliability Enhancement of Navier-Stokes Codes Through Convergence Enhancement, NASA Propulsion Engineering Research Center Fifth Annual Symposium, University Park, PA (September 8-9, 1993) 181-186.
- [13] K.-Y. Choi and G.S. Dulikravich, Convergence Acceleration of Iterative Algorithms Using a Sensitivity-Based Minimum Residual (SBMR) Method, 12th Workshop for CFD Applications in Rocket Propulsion, NASA MSFC, Huntsville, AL (April 19-21, 1994)
- [14] K.-Y. Choi and G.S. Dulikravich, Global Sensitivity-Based Minimal Residual (GSBMR) Method for Acceleration of Iterative Convergence Rates, Proceedings of the 14th International Conference on Numerical Fluid Dynamics, Bangalore, India (July 11-15, 1994)
- [15] K.-Y. Choi and G.S. Dulikravich, Convergence Rate Enhancement of Navier-Stokes Codes on Clustered Grids, 6th Annual Symposium of the Penn State - NASA Propulsion Engineering Research Center, NASA Lewis Research Center, Cleveland, OH (September 13-14, 1994)
- [16] K.-Y. Choi, Sensitivity-Based Minimum Residual Methods for Convergence Acceleration of Iterative Algorithms, Ph. D. dissertation, Department of Aerospace Engineering, The Pennsylvania State University (December 1994)
- [17] A.J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, Journal of Computational Physics, 2, (1967) 12-26.
- [18] K.W. Thomson, Time-dependent Boundary Conditions for Hyperbolic Systems, II, Journal of Computational Physics, Vol. 89, (1990) 439-461.
- [19] G.S. Dulikravich, V. Ahuja and S. Lee, Modeling Three-Dimensional Solidification With Magnetic Fields and Reduced Gravity, International Journal of Heat and Mass Transfer, 37, (5), (1994) 837-853.
- [20] D.A. Anderson, J.C. Tannehill and R.H. Fletcher, Computational Fluid Mechanics and Heat Transfer (McGraw-Hill Book Company, New York, New York, 1984) 249-250.
- [21] B.S. Baldwin and H. Lomax, Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows, AIAA Paper 78-257, Huntsville, AL, January 16-18, 1978.

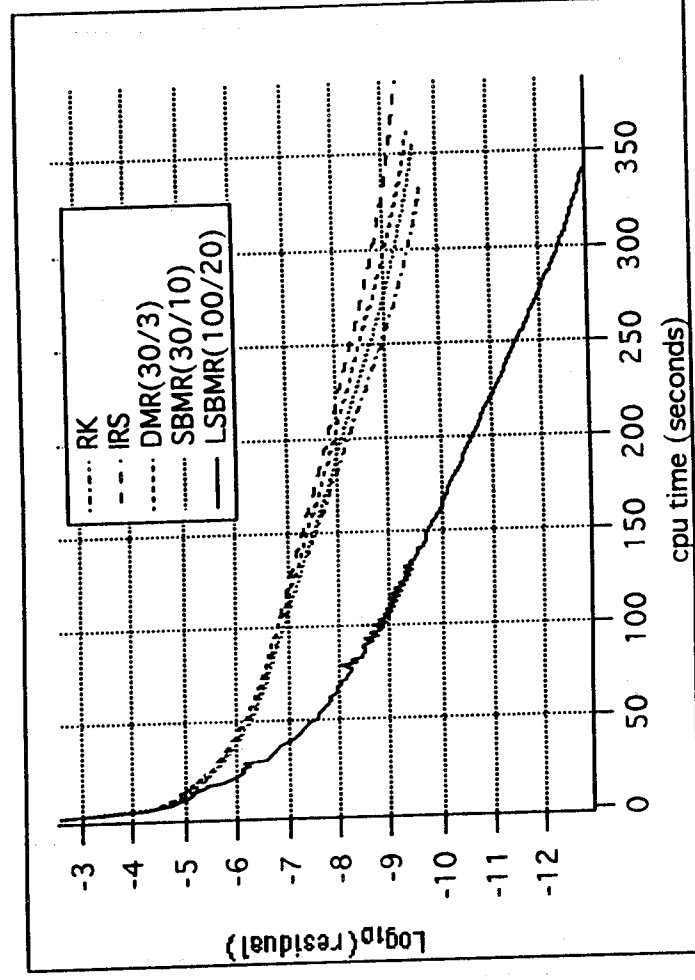
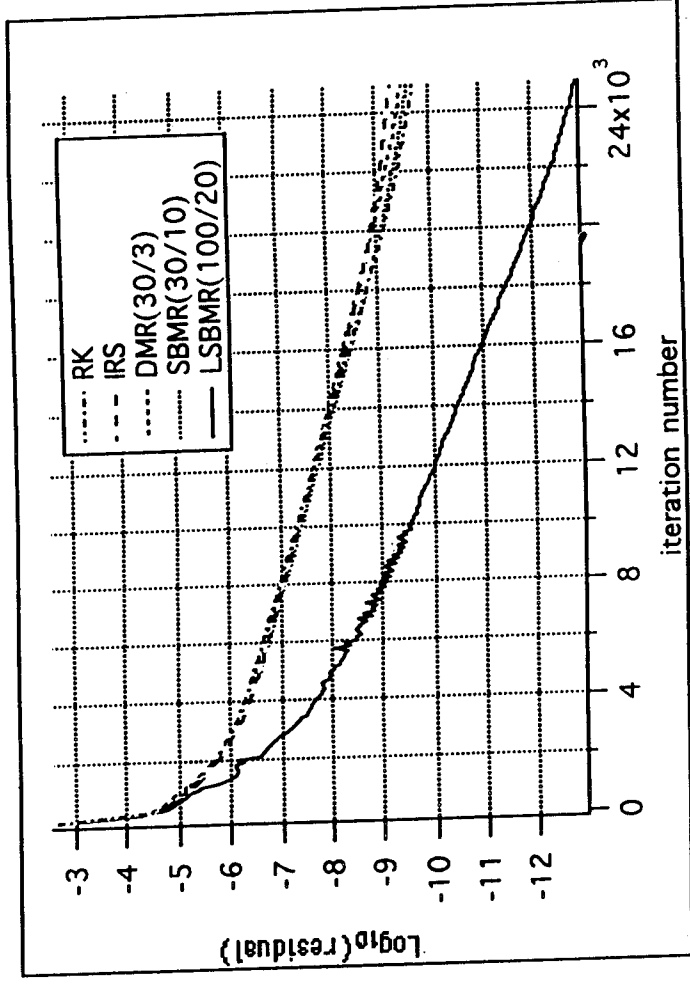
LIST OF FIGURE CAPTIONS

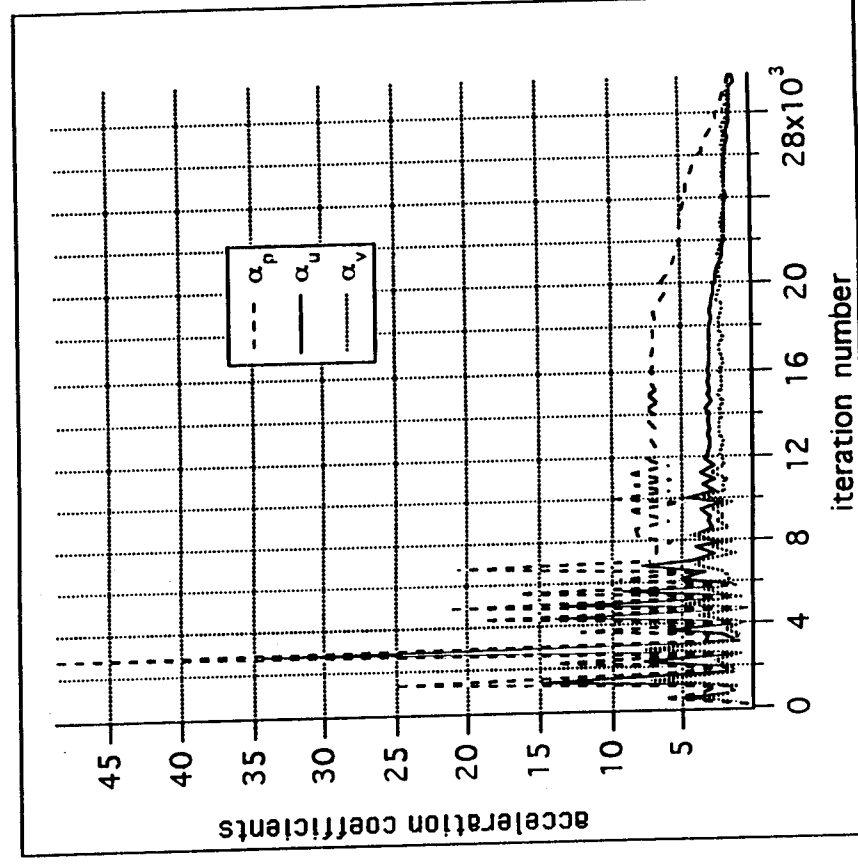
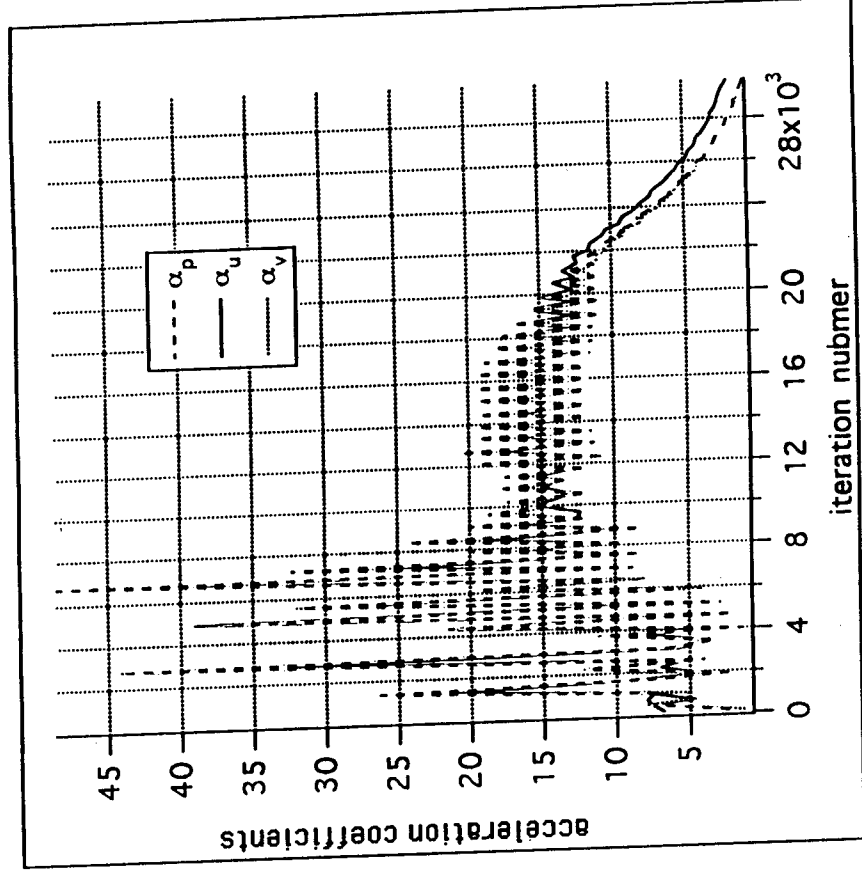
- Figure 1 Convergence histories for a 2-D straight channel flow ($L/H = 5$, $Re = 1600$, 60×60 grid cells, $AR_{\max} = 10$, $\beta = 5$, $CFL = 2.8$, $\sigma = 0.4$)
- Figure 2 Convergence histories for a 2-D straight channel flow ($L/H = 5$, $Re = 1600$, 60×60 grid cells, $AR_{\max} = 100$, $\beta = 5$, $CFL = 2.8$, $\sigma = 0.4$)
- Figure 3 SBMR method: Relative errors in pressure drop (a) and axial velocity component (b) at different x-locations: i = 1 inlet; i = 61 exit ($Re = 1600$, $AR_{\max} = 100$, $Freq = 30$, $n = 10$)
- Figure 4 Convergence histories for a 2-D straight channel flow ($L/H = 5$, $Re = 1600$, 60×60 grid cells, $AR_{\max} = 10,000$, $\beta = 5$, $CFL = 2.8$, $\sigma = 0.4$)
- Figure 5 LSBMR method: Acceleration coefficients for a 2-D straight channel flow at a) $j = 2$, and b) $j = 31$ ($AR_{\max} = 10,000$, $Re = 1600$, $Freq = 100$, $n = 20$)
- Figure 6 Convergence histories for a 2-D straight channel flow ($L/H = 10$, $Re = 1.6 \times 10^6$, 60×120 grid cells, $AR_{\max} = 1,000$, $\beta = 5$, $CFL = 2.0$, $\sigma = 0.4$)
- Figure 7 Convergence histories for a 2-D straight channel flow ($L/H = 10$, $Re = 1.6 \times 10^6$, 60×120 grid cells, $AR_{\max} = 10,000$, $\beta = 5$, $CFL = 2.0$, $\sigma = 0.4$)
- Figure 8 Isobars (a) and velocity vector field (b) computed for a U-shaped channel flow ($Re = 200$, $AR_{\max} = 200$, 129×80 grid cells)
- Figure 9 Convergence histories for a U-shaped channel flow ($Re = 200$, $AR_{\max} = 40$, $\beta = 10$, $CFL = 2.8$, $\sigma = 0.4$, 129×80 grid cells)
- Figure 10 Convergence histories for a U-shaped channel flow ($Re = 200$, $AR_{\max} = 200$, $\beta = 10$, $CFL = 2.8$, $\sigma = 0.4$, 129×80 grid cells)

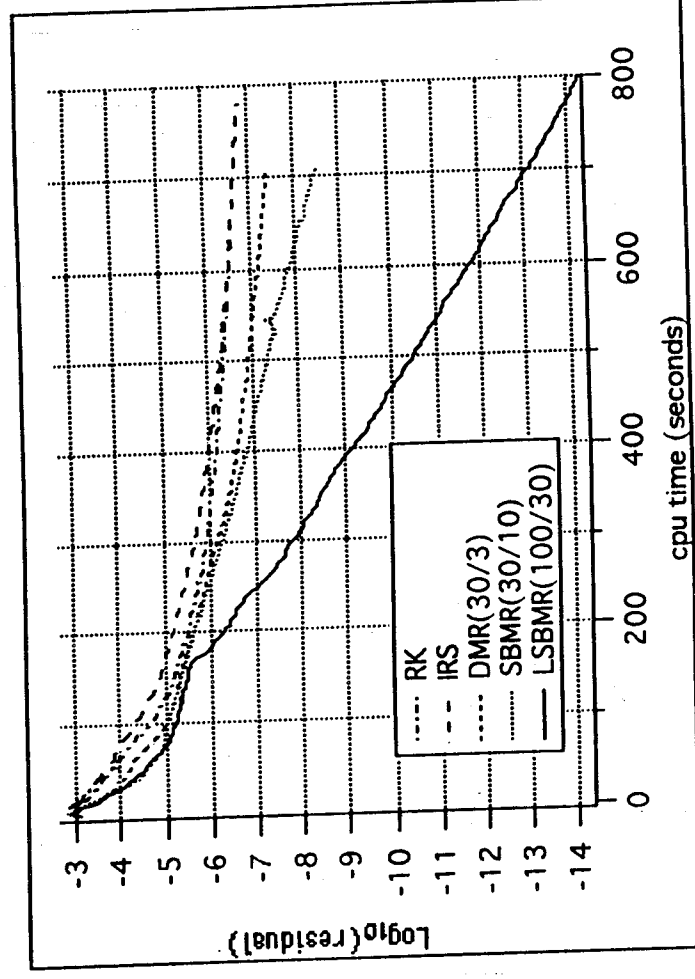
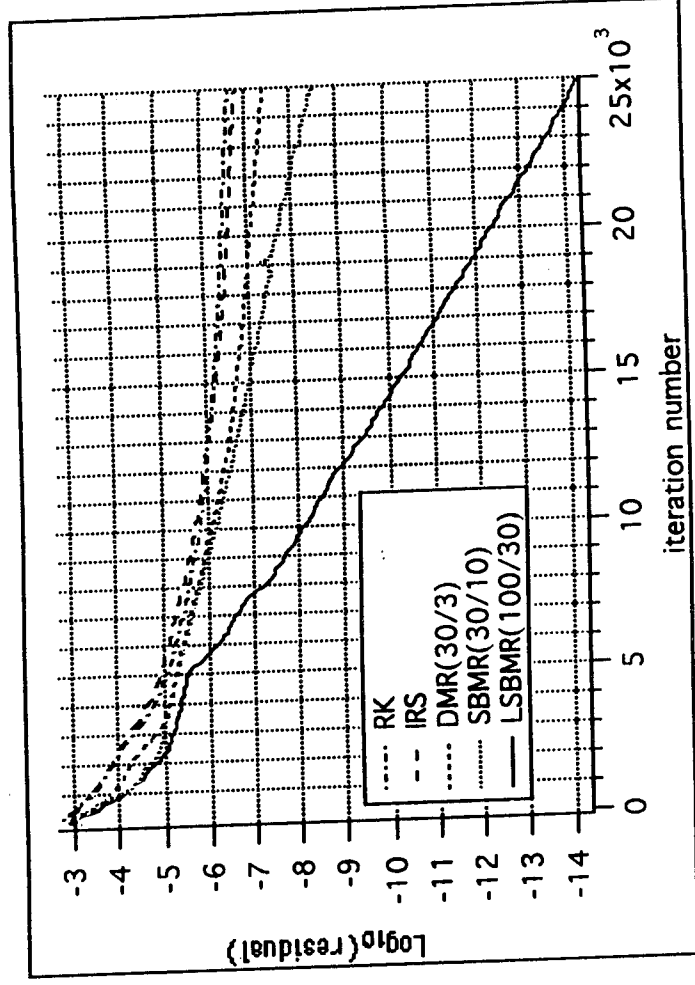


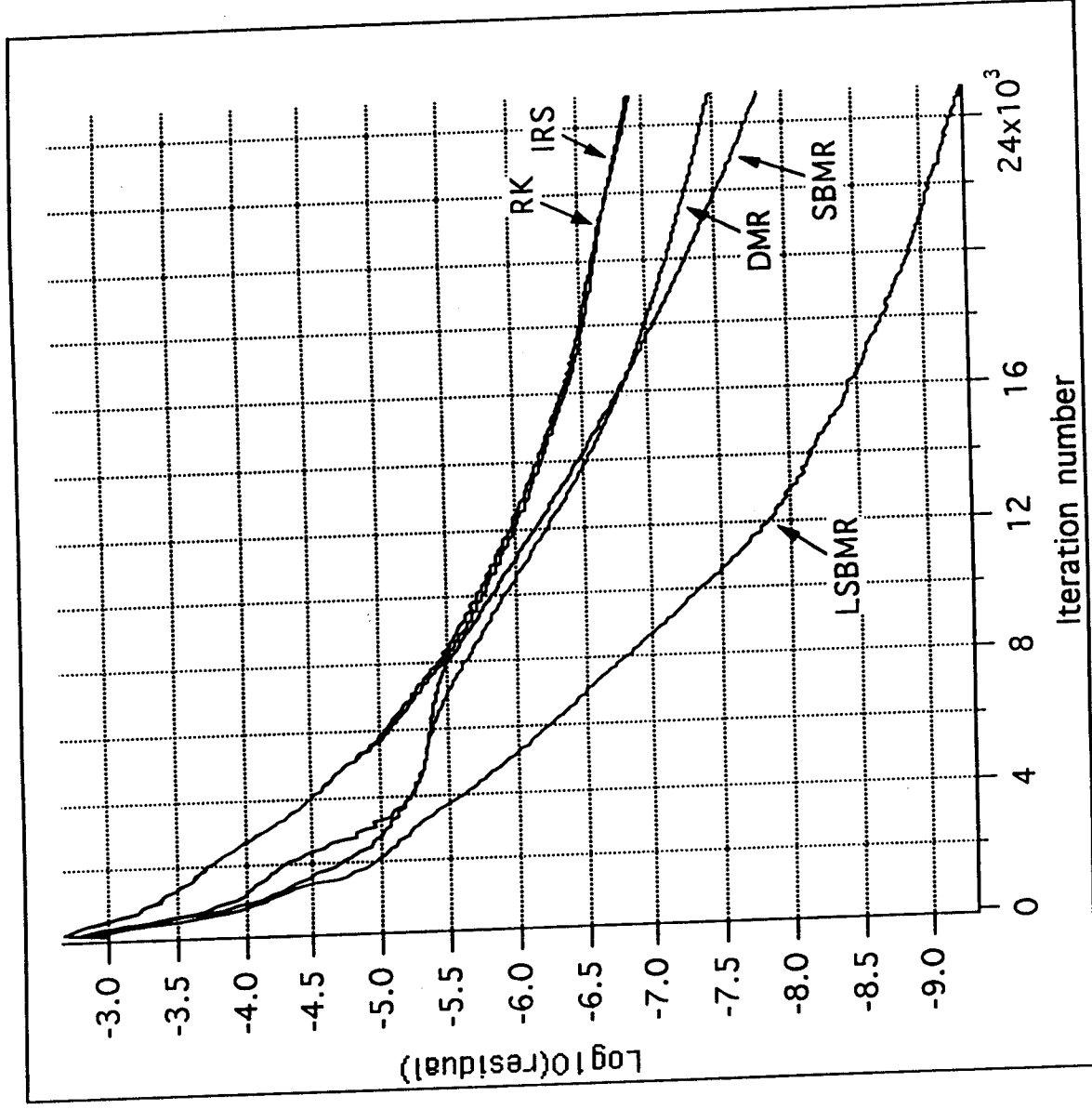


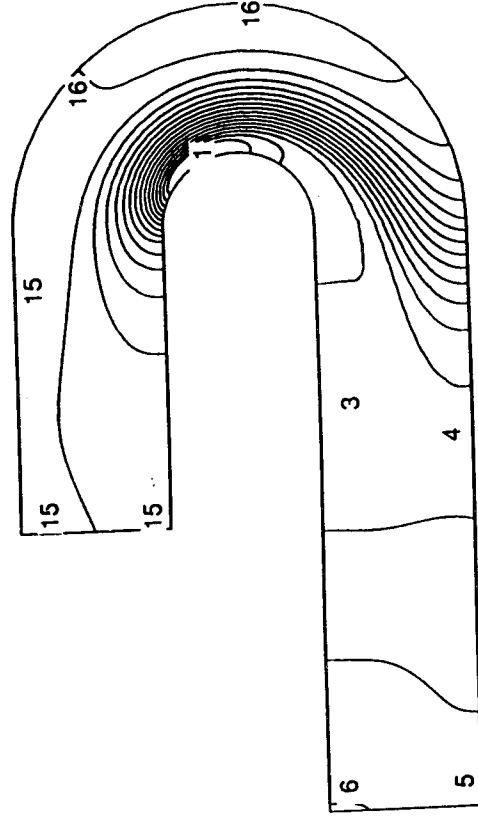






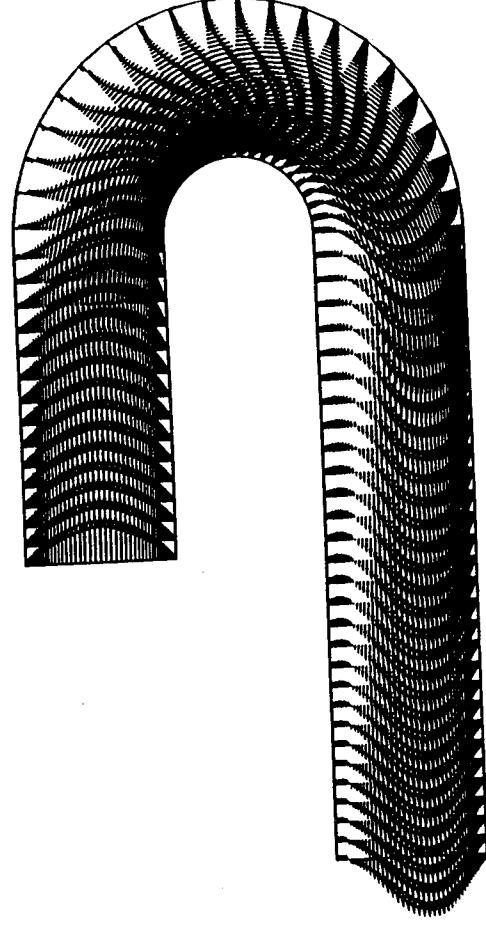




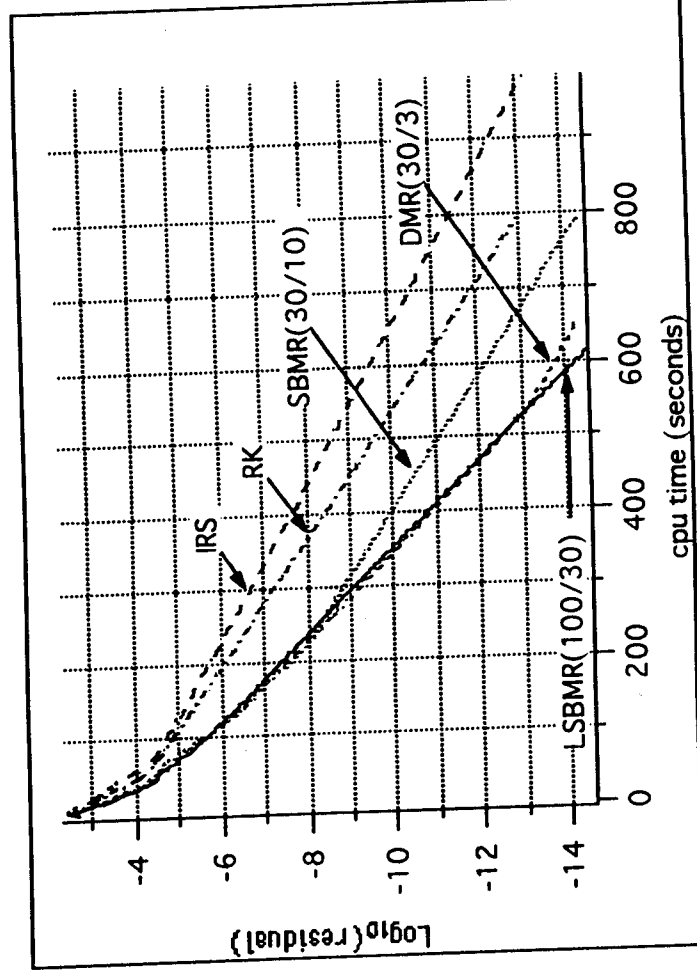
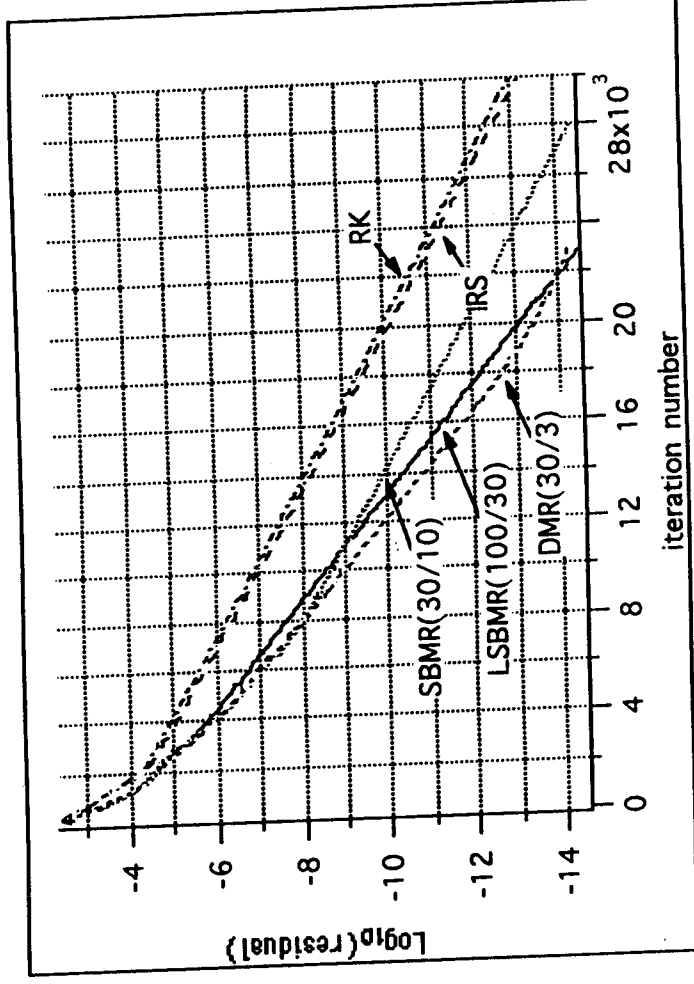


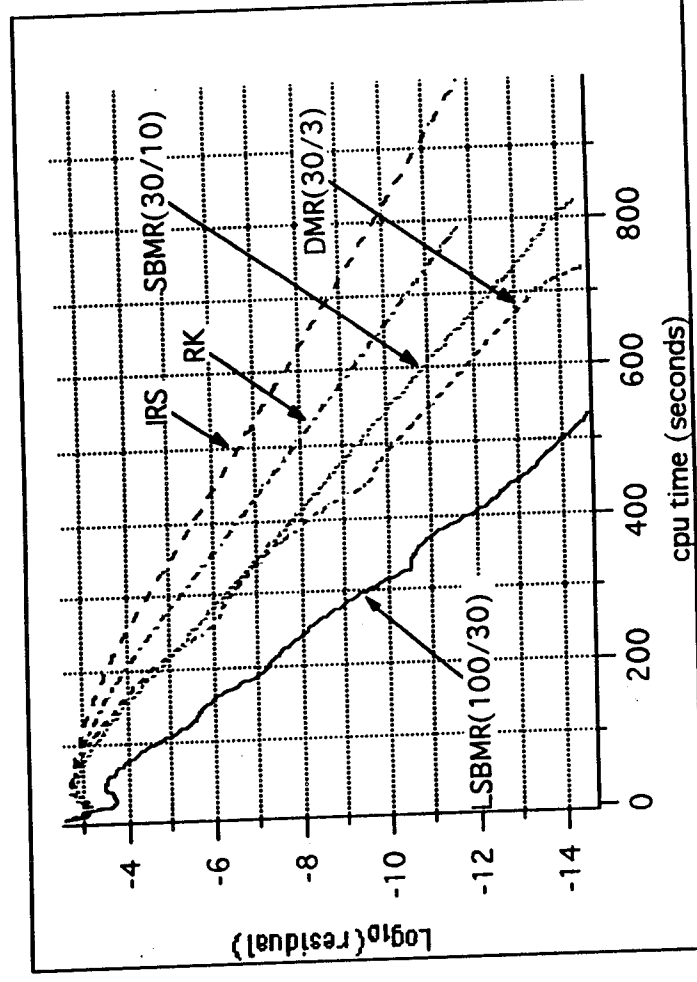
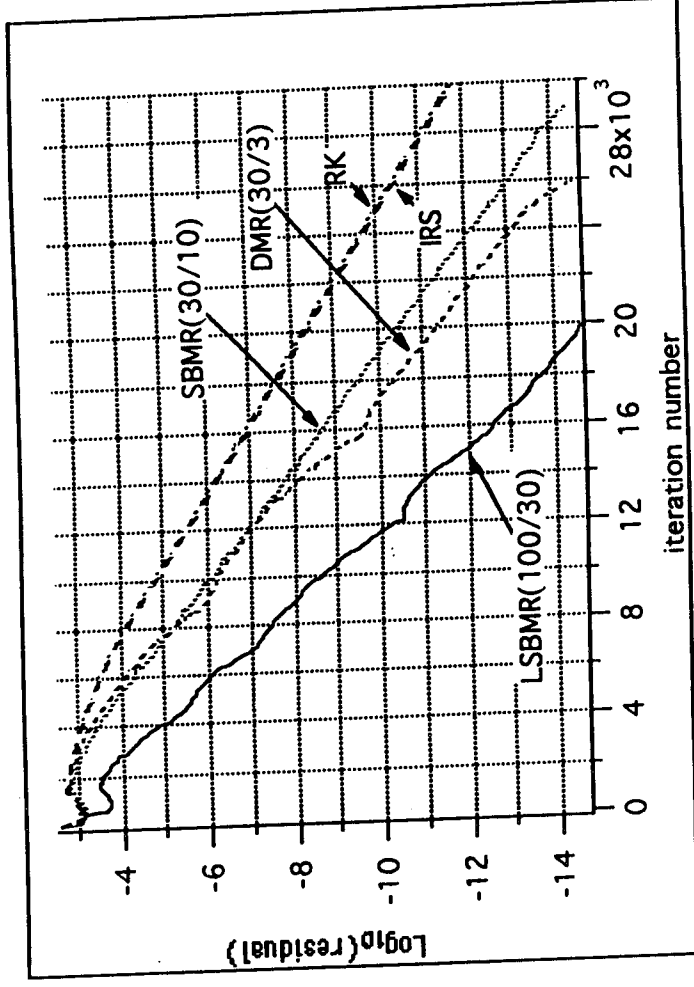
a

1	-1.4000
2	-1.3000
3	-1.2000
4	-1.1000
5	-1.0000
6	-0.9000
7	-0.8000
8	-0.7000
9	-0.6000
10	-0.5000
11	-0.4000
12	-0.3000
13	-0.2000
14	-0.1000
15	0.0000
16	0.1000
17	0.2000



b





Acceleration of Iterative Algorithms on Highly Clustered Grids

Final Report

K.-Y. Choi and G. S. Dulikravich

ACCELERATION OF ITERATIVE ALGORITHMS ON HIGHLY CLUSTERED GRIDS

Kwang-Yoon Choi and George S. Dulikravich

Graduate Research Assistant Associate Professor

Department of Aerospace Engineering

The Pennsylvania State University

University Park, Pennsylvania 16802

Running title: Sensitivity-Based Acceleration Methods

Key words:

numerical methods

acceleration schemes

fast algorithms

sensitivity analysis

All correspondence should be sent to:

Prof. George S. Dulikravich

Department of Aerospace Engineering

233 Hammond Bldg.

The Pennsylvania State University

University Park, PA 16802, USA

(814) 863-0134 phone (814) 865-7092 FAX gsd@ecl.psu.edu

Summary

A family of new methods to accelerate the convergence rate of iterative algorithms for obtaining a steady-state solution as an asymptotic limit of an unsteady second-order partial differential equation or a system of such equations has been developed with an assumption that a central differencing has been used for spatial discretization. The new acceleration methods are based on the sensitivity of the future residual at every grid point to the change in the solution vector components at the neighboring grid points used in the local discretization approximation. The acceleration parameters introduced in the methods have been optimized with the objective to minimize the future global residual. The new sensitivity-based methods have been applied to finite difference codes for two- and three-dimensional laminar incompressible flow Navier-Stokes equations, two-dimensional turbulent incompressible flow Navier-Stokes equations and two-dimensional compressible flow Euler equations. The new sensitivity-based acceleration methods demonstrated superior performance in all test cases that involved severe grid clustering and grid non-orthogonality and included laminar and turbulent flows with closed and open flow separation.

I. Introduction

To accurately and reliably resolve details of field problems where variables experience very sharp gradients, it is necessary to use strong local grid clustering. Nevertheless, the performance of both implicit and explicit iterative algorithms for the integration of systems of nonlinear partial differential equations suffers from a slow convergence on such highly-clustered grids, because local time steps are related to the local grid spacing. Several attempts have been made to accelerate the convergence of such algorithms using local time-stepping^{1,2}, implicit residual smoothing³, enthalpy damping¹ and multi-gridding⁴⁻⁷. None of these schemes was capable of accelerating the convergence rates on highly clustered grids. Despite the multigrid method's superior capability of effectively reducing low and high frequency errors yielding impressive convergence rates, its

3
efficiency is significantly reduced on a highly clustered grid⁵⁻⁷ and it is difficult to implement reliably⁸. The preconditioning methods⁸⁻¹⁰, although very powerful in alleviating the slow convergence associated with a stiff system for solving the low Mach number compressible flow equations, have not been shown to perform universally well on highly clustered grids.

Optimum extrapolation techniques represent an entirely different type of convergence acceleration algorithms that belong to the general class of Krylov subspace methods. One such technique is the Minimal Residual Method (MRM) which uses an equal optimal weight for the corrections to every equation¹ in a system of partial differential equations¹² that is solved. The General Nonlinear Minimal Residual (GNLMR) method^{13 14} allows for an identical sequence of optimized relaxation parameters to be applied to each component of the solution vector in a system that is solved. The Distributed Minimal Residual (DMR) method¹⁵⁻¹⁷ allows each component of the solution vector in a system of equations to have its own sequence of optimized relaxation parameters. The application of the DMR method requires that solutions from three or four consecutive iterations be stored, which represents an improvement over the General Minimal Residual (GMRES) method¹⁸ which requires storing a large number of solutions. None of these methods was found to accelerate the iterative convergence rates on highly clustered grids.

The objective of the present study is to develop a numerical method that is capable of accelerating the convergence of arbitrary iterative algorithms for integrating a general system of hyperbolic partial differential equations and to investigate the performance of the newly developed scheme on a variety of fluid dynamic problems with a special emphasis on the acceleration of convergence on highly clustered grids. Four-stage Runge-Kutta time integration method with local time-stepping will be used as a basic iterative algorithm and spatial derivatives will be approximated by a second order accurate central differencing throughout the study. Characteristic boundary conditions will be used at the inlet and non-reflecting boundary conditions will be used at the exit boundary of the flow domain. We will be seeking accurate steady-state solutions only.

II. Global SBMR Method

A system of unsteady partial differential equations can be written as $\mathbf{R} = -\frac{\partial \mathbf{Q}}{\partial t} = L(\mathbf{Q})$ where $\mathbf{Q}$ is the solution vector, t is the time, L is the differential operator and $\mathbf{R}$ is the residual vector. Conventional iterative algorithms update $\mathbf{Q}$ by calculating the amount of corrections, $\Delta \mathbf{Q}$, without optimizing their influence on the future residual. The basic idea of our Sensitivity-Based Minimum Residual (SBMR) method¹⁹⁻²³ is that the change of the residual at a grid point depends on the changes of $\mathbf{Q}$ at the neighboring grid points used in the local finite difference approximation. The sensitivities can be determined by taking partial derivatives of the finite difference approximation of $\mathbf{R}$ with respect to each component of $\mathbf{Q}$. This information is utilized to effectively extrapolate $\mathbf{Q}$ so as to minimize the future residual. The rate at which the residual vector components, R_r ($r = 1, \dots, r_{\max}$: number of equations in the system), change with the solution vector components, Q_{ms} , ($m = 1, \dots, M$: number of unknowns; $s = 1, \dots, S$: number of surrounding grid points directly involved in

the local discretization scheme) is $\frac{\partial R_r}{\partial Q_{ms}}$. In formulating the global SBMR method for a three-dimensional problem when using central differencing, there will be nineteen grid points involved in the local sensitivity calculation compared to nine grid points for a two-dimensional case. Suppose we know the solution vectors $\mathbf{Q}^t$ and $\mathbf{Q}^{t+n}$ at iteration levels t and $t+n$, respectively. Here, n is the number of regular iteration steps performed by the original non-accelerated algorithm. Then $\Delta \mathbf{Q}$ between the two iteration levels is given as $\mathbf{Q}^{t+n} = \mathbf{Q}^t + \Delta \mathbf{Q}$. Using the first two terms of a Taylor series expansion in the artificial time direction, the residual for each of the equations in the system after n iterations is

$$R_r^{t+n} = R_r^t + \sum_m^M \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \quad (1)$$

Here, two subscripts were used in components of ΔQ . The first subscript, m , is the identifier for the solution variable and the second, s , is for the indexing of the surrounding grid points. Notice that the total number of equations in the system is the same as the total number of unknown components of Q , that is, $\text{rmax} = M$. If we introduce convergence rate acceleration coefficients $\alpha_1, \alpha_2, \dots, \alpha_M$ multiplying corrections $\Delta Q_{1,s}, \Delta Q_{2,s}, \dots, \Delta Q_{M,s}$ respectively, the future solution vector component, $Q_{m,s}^{(t+n)+1}$, can be extrapolated as

$$Q_{m,s}^{(t+n)+1} = Q_{m,s}^t + \alpha_m \Delta Q_{m,s} \quad (2)$$

For the global SBMR method, each of the α 's is assumed to have the same value over the entire domain, D . Subsequently, the future residual at the iteration level $(t+n)+1$ can be approximated by

$$R_r^{(t+n)+1} = R_r^t + \sum_m^M \sum_s^S \left[\frac{\partial R_r^t}{\partial Q_{m,s}} \alpha_m \Delta Q_{m,s} \right] \quad (3)$$

or, since α 's are assumed constant at every grid point for the global SBMR method, as

$$R_r^{(t+n)+1} = R_r^t + \sum_m^M \left[\alpha_m \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \right] \quad (4)$$

Subtracting (1) from (4) yields

$$R_r^{(t+n)+1} = R_r^{t+n} + \sum_m^M (\alpha_m - 1) a_{r,m} \quad (5)$$

where $a_{r,m} = \sum_s \frac{\partial R_r^s}{\partial Q_{m,s}}$. The optimum α 's are determined such that the sum of the L-2 norm of

the future residuals over the entire domain D will be minimized. In general, this means that

$$\sum_D \sum_r \frac{\partial (R_r^{(t+n+1)})^2}{\partial \alpha_m} = 2 \sum_D \sum_r R_r^{(t+n+1)} \frac{\partial R_r^{(t+n+1)}}{\partial \alpha_m} = 0 \quad (6)$$

for $m = 1, 2, \dots, M$. With the help of (5), the system (6) becomes

$$\begin{aligned} \sum_D \left[\sum_r \left\{ R_r^{t+n} + \sum_m^M a_{r,m} (\alpha_m - 1) \right\} a_{r,1} \right] &= 0 \\ \sum_D \left[\sum_r \left\{ R_r^{t+n} + \sum_m^M a_{r,m} (\alpha_m - 1) \right\} a_{r,2} \right] &= 0 \\ &\vdots \\ \sum_D \left[\sum_r \left\{ R_r^{t+n} + \sum_m^M a_{r,m} (\alpha_m - 1) \right\} a_{r,M} \right] &= 0 \end{aligned} \quad (7)$$

In equation (7), R's and a's are known from the preceding iteration levels. Since each α is assumed to have the same value over the entire computational domain, equation (7) gives a system of M simultaneous algebraic equations for M optimum $\alpha_1, \alpha_2, \dots, \alpha_M$

$$\begin{aligned} \left[\sum_D \left(\sum_r a_{r,1} a_{r,1} \right) \right] (\alpha_1 - 1) + \left[\sum_D \left(\sum_r a_{r,2} a_{r,1} \right) \right] (\alpha_2 - 1) + \dots + \left[\sum_D \left(\sum_r a_{r,M} a_{r,1} \right) \right] (\alpha_M - 1) &= - \sum_D \left(\sum_r R_r^{t+n} a_{r,1} \right) \\ \left[\sum_D \left(\sum_r a_{r,1} a_{r,2} \right) \right] (\alpha_1 - 1) + \left[\sum_D \left(\sum_r a_{r,2} a_{r,2} \right) \right] (\alpha_2 - 1) + \dots + \left[\sum_D \left(\sum_r a_{r,M} a_{r,2} \right) \right] (\alpha_M - 1) &= - \sum_D \left(\sum_r R_r^{t+n} a_{r,2} \right) \\ &\vdots \\ \left[\sum_D \left(\sum_r a_{r,1} a_{r,M} \right) \right] (\alpha_1 - 1) + \left[\sum_D \left(\sum_r a_{r,2} a_{r,M} \right) \right] (\alpha_2 - 1) + \dots + \left[\sum_D \left(\sum_r a_{r,M} a_{r,M} \right) \right] (\alpha_M - 1) &= - \sum_D \left(\sum_r R_r^{t+n} a_{r,M} \right) \end{aligned} \quad (8)$$

For the general case of a system composed of M partial differential equations with M unknowns, the system (8) will become a full $M \times M$ symmetric matrix for M unknown optimum α 's. For the rest of this study, the global SBMR method^{19 20} will be called SBMR method to differentiate it from its descendants, such as line SBMR (LSBMR), plane SBMR (PSBMR) or alternating direction plane SBMR (APSBMR) methods, that will be designed for special purposes.

III. Line SBMR (LSBMR) Method

It is plausible that for non-uniform computational grids and rapidly varying dependent variables, optimum α 's should not necessarily be the same over the whole computational domain. A modification of the SBMR method called Line Sensitivity-Based Minimal Residual (LSBMR) method was developed to allow α 's to have different values from one grid line to another. The LSBMR formulation²⁰⁻²³ will be explained using the two-dimensional, incompressible flow Navier-Stokes equations as an example. The system has three equations ($rmax = 3$) and three unknowns ($M = 3$), that are pressure, p , and the local velocity vector Cartesian components, u and v . The acceleration coefficients for those unknowns are α_p , α_u and α_v respectively. If the grid lines are clustered in the j -direction, then each $j = \text{constant}$ grid line will be assigned its own set of constant α 's. The residual at a grid point (i,j) incorporates α 's at the neighboring grid points plus the point itself. For the given Navier-Stokes system, this yields

$$R_r^{(i+j)+1} = R_r^i + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_r^i}{\partial p_{s,j-1}} \Delta p_{s,j-1} \alpha_p^{j-1} + \frac{\partial R_r^i}{\partial p_{s,j}} \Delta p_{s,j} \alpha_p^j + \frac{\partial R_r^i}{\partial p_{s,j+1}} \Delta p_{s,j+1} \alpha_p^{j+1} \right]$$

$$+ \sum_{s=i-1}^{i+1} \left[\frac{\partial R_r^i}{\partial u_{s,j-1}} \Delta u_{s,j-1} \alpha_u^{j-1} + \frac{\partial R_r^i}{\partial u_{s,j}} \Delta u_{s,j} \alpha_u^j + \frac{\partial R_r^i}{\partial u_{s,j+1}} \Delta u_{s,j+1} \alpha_u^{j+1} \right]$$

$$+ \sum_{s=i-1}^{i+1} \left[\frac{\partial R_r^t}{\partial v_{sj-1}} \Delta v_{sj-1} \alpha_v^{j-1} + \frac{\partial R_r^t}{\partial v_{sj}} \Delta v_{sj} \alpha_v^j + \frac{\partial R_r^t}{\partial v_{sj+1}} \Delta v_{sj+1} \alpha_v^{j+1} \right] \quad (9)$$

for $j = 1, 2, \dots, j_{\max}$. For a given $j = \text{constant}$ grid line, when using central differencing in the j -direction, α^j appears only in $R_m^{(t+n)+1}(i, j-1)$, $R_m^{(t+n)+1}(i, j)$ and $R_m^{(t+n)+1}(i, j+1)$. Therefore, we can see that the summation over the entire domain leaves the terms only with α^{j-2} , α^{j-1} , α^j , α^{j+1} and α^{j+2} summed along each $j = \text{constant}$ line. Thus, on each $j = \text{constant}$ line, M values of constant α 's are determined so as to minimize the L-2 norm of the future global residual. The minimization results in $(j_{\max}) \times M$ algebraic equations for the same number of unknown optimum α 's.

$$\begin{aligned} & 2 \sum_{i=1}^{i_{\max}} \sum_{r=1}^3 \left[R_r^{(t+n)+1}(i, j-1) \frac{\partial R_r^{(t+n)+1}(i, j-1)}{\partial \alpha_m^j} \right] \\ & + 2 \sum_{i=1}^{i_{\max}} \sum_{r=1}^3 \left[R_r^{(t+n)+1}(i, j) \frac{\partial R_r^{(t+n)+1}(i, j)}{\partial \alpha_m^j} \right] \\ & + 2 \sum_{i=1}^{i_{\max}} \sum_{r=1}^3 \left[R_r^{(t+n)+1}(i, j+1) \frac{\partial R_r^{(t+n)+1}(i, j+1)}{\partial \alpha_m^j} \right] = 0 \end{aligned} \quad (10)$$

This should be repeated for $j = 1, 2, \dots, j_{\max}$ and $m = p, u, v$. In this example the simultaneous system of equations (10) yields a block penta-diagonal matrix equation for $(j_{\max}) \times 3$ optimum α 's where each block is a 3×3 matrix. In the general case of a two-dimensional system having M partial differential equations, the block penta-diagonal system (10) will have blocks of size $M \times M$. When $m = p$, the terms inside the summation sign of equation (10) can be written as follows.

$$R_r^{(t+m-1)}(i,j-1) \frac{\partial R_r^{(t+m+1)}(i,j-1)}{\partial \alpha_p^j} =$$

$$\left[\begin{aligned} & R_r^t(i,j-1) + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial p_{sj-2}} \Delta p_{sj-2}^{j-2} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial p_{sj-1}} \Delta p_{sj-1}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial p_{sj}} \Delta p_{sj}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial p_{sj}} \Delta p_{sj}^j \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial u_{sj-2}} \Delta u_{sj-2}^{j-2} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial u_{sj-1}} \Delta u_{sj-1}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial u_{sj}} \Delta u_{sj}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial u_{sj}} \Delta u_{sj}^j \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial v_{sj-2}} \Delta v_{sj-2}^{j-2} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial v_{sj-1}} \Delta v_{sj-1}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial v_{sj}} \Delta v_{sj}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial v_{sj}} \Delta v_{sj}^j \end{aligned} \right] \times \left[\sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j-1)}{\partial p_{sj}} \Delta p_{sj} \right]$$

$$R_r^{(t+m-1)}(i,j) \frac{\partial R_r^{(t+m+1)}(i,j)}{\partial \alpha_p^j} =$$

$$\left[\begin{aligned} & R_r^t(i,j) + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial p_{sj-1}} \Delta p_{sj-1}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial p_{sj}} \Delta p_{sj}^j + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial p_{sj+1}} \Delta p_{sj+1}^{j+1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial p_{sj+1}} \Delta p_{sj+1}^j \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial u_{sj-1}} \Delta u_{sj-1}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial u_{sj}} \Delta u_{sj}^j + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial u_{sj+1}} \Delta u_{sj+1}^{j+1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial u_{sj+1}} \Delta u_{sj+1}^j \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial v_{sj-1}} \Delta v_{sj-1}^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial v_{sj}} \Delta v_{sj}^j + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial v_{sj+1}} \Delta v_{sj+1}^{j+1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial v_{sj+1}} \Delta v_{sj+1}^j \end{aligned} \right] \times \left[\sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i,j)}{\partial p_{sj}} \Delta p_{sj} \right]$$

with a similar expression for $(i,j+1)$ grid line and similar expressions for $m = u$ and $m = v$.

Substituting these equations into equation (10) and collecting terms with α_p , α_u and α_v , we can construct the first row of each block in the block penta-diagonal matrix. Other expressions similar to these for $m = u$ and $m = v$ give the second and third row of each block respectively. The overall structure of the block penta-diagonal matrix is

$$\begin{bmatrix} [C^j] & [D^j] & [E^j] & & \\ & \ddots & & & \\ \dots & [A^{j-1}] & [B^{j-1}] & [C^{j-1}] & [D^{j-1}] & [E^{j-1}] \\ & [A^j] & [B^j] & [C^j] & [D^j] & [E^j] \\ & [A^{j+1}] & [B^{j+1}] & [C^{j+1}] & [D^{j+1}] & [E^{j+1}] & \dots \\ & & \ddots & & & \\ & & & [A^{j\max}] & [B^{j\max}] & [C^{j\max}] \end{bmatrix} = \begin{bmatrix} \{\alpha^1\} \\ \vdots \\ \{\alpha^{j-2}\} \\ \{\alpha^{j-1}\} \\ \{\alpha^j\} \\ \{\alpha^{j+1}\} \\ \{\alpha^{j+2}\} \\ \vdots \\ \{\alpha^{j\max}\} \end{bmatrix} = \begin{bmatrix} \{F^1\} \\ \vdots \\ \{F^{j-2}\} \\ \{F^{j-1}\} \\ \{F^j\} \\ \{F^{j+1}\} \\ \{F^{j+2}\} \\ \vdots \\ \{F^{j\max}\} \end{bmatrix} \quad (13)$$

where $\{\alpha^j\} = \begin{bmatrix} \alpha_p^j \\ \alpha_u^j \\ \alpha_v^j \end{bmatrix}$ and $\{F^j\}$ is the vector of terms that do not contain α 's. The row elements of the blocks $[A], [B], [C], [D]$ and $[E]$ are coefficients of $\alpha^{j-2}, \alpha^{j-1}, \alpha^j, \alpha^{j+1}$ and α^{j+2} , respectively. Each column of the blocks $[A], [B], [C], [D]$ and $[E]$ corresponds to α_p, α_u and α_v , respectively.

Application of the LSBMR method to a three-dimensional problem would result in a large, unstructured sparse block matrix rather than a block penta-diagonal matrix as for the two-dimensional LSBMR. For this reason it is simpler to use the Plane SBMR (PSBMR) and Alternating Direction SBMR (APSBMR) methods for three-dimensional problems. The PSBMR method assumes that each grid plane has its own set of α 's. Suppose j = constant grid plane has a fixed set of α 's. Then, the PSBMR method yields

$$2 \sum_{i=1}^{i\max} \sum_{k=1}^{k\max} R_r^{(t+n)+1}(i, j-1, k) \frac{\partial R_r^{(t+n)+1}(i, j-1, k)}{\partial \alpha_m^j} \Bigg]$$

$$\begin{aligned}
& + 2 \sum_{i=1}^{\text{imax}} \sum_{k=1}^{\text{kmax}} \sum_{r=1}^{\text{rmax}} \left[R_r^{(t+n)+1}(i, j, k) \frac{\partial R_r^{(t+n)+1}(i, j, k)}{\partial \alpha_m^j} \right] \\
& + 2 \sum_{i=1}^{\text{imax}} \sum_{k=1}^{\text{kmax}} \sum_{r=1}^{\text{rmax}} \left[R_r^{(t+n)+1}(i, j+1, k) \frac{\partial R_r^{(t+n)+1}(i, j+1, k)}{\partial \alpha_m^j} \right] = 0
\end{aligned} \tag{14}$$

for $m = 1, 2, \dots, M$. A variation of the PSBMR method, called APSBMR (Alternating direction Plane SBMR) method can be used by applying the PSBMR method alternately on $j = \text{constant}$ and on $k = \text{constant}$ planes. This could be useful when the grids are clustered in both j and k directions.

IV. Computational Results

A. Two-Dimensional Flows

Convergence histories of the basic four-stage RK scheme with local time-stepping, and four acceleration methods (Implicit Residual Smoothing¹ (IRS), DMR, SBMR and LSBMR) were compared for a steady, incompressible, viscous flow in a two-dimensional straight channel and a U-shaped turn-around channel. The inlet pressure was iteratively computed by enforcing characteristic boundary conditions, while at the exit boundary we used non-reflecting boundary conditions^{24 25} since there was a recirculating flow at the exit of the U-shape channel. No artificial viscosity was used in any of the test cases. The general algorithm used Chorin's artificial compressibility method²⁶. The Courant-Friedrichs-Levy number and von Neumann number had maximum possible values ($\text{CFL} = 2.8$ and $\sigma = 0.4$) in all test cases discussed in this paper in order to achieve maximum performance of the basic RK scheme.

For laminar flow in a straight channel with mild symmetric grid clustering towards the walls (maximum cell aspect ratio was $AR_{\max} = (\Delta x / \Delta y)_{\max} = 100$) the LSBMR method converged faster than the SBMR method (Fig. 1). This was an expected result, since the LSBMR method allows different optimum α 's in the clustered grid region (Fig. 2) rather than enforcing fixed acceleration parameters over the whole computational domain as required by the SBMR. The computational results compared^{22 23} with analytical solutions to within machine accuracy. Convergence histories versus computational time consistently mirror the convergence histories versus number of iterations. Both, SBMR and LSBMR methods, outperformed the basic RK scheme over a range of grid non-orthogonality angles (Fig. 3). Sensitivity of the two methods to the user-specified frequency of application and to the number of iterations apart appearing in the equation, are depicted in Fig. 4 and Fig. 5, respectively. Further increase in grid clustering to $AR_{\max} = 10,000$ clearly slows down the convergence rates of all methods tested (Fig. 6). However, the LSBMR method maintains a considerably faster convergence especially when combined with the simple Time-Step Scaling²² (TSS) method where the local time step for the continuity equation is scaled such that $J\Delta\tau$ in the clustered regions has the same order of magnitude as that in the coarse grid regions. Here, J is the determinant of the local Jacobian transformation matrix between the physical (x,y) and the curvilinear non-orthogonal boundary-conforming computational (ξ,η) coordinate system, while $\Delta\tau$ is the local time step computed using standard formulation²⁷. The convergence acceleration schemes were then tested for a turbulent flow²⁸ (Reynolds number $Re = 1.6$ million) in a straight channel. Convergence histories for $AR_{\max} = 1,000$ demonstrate (Fig. 7) that the LSBMR method consistently outperformed other acceleration schemes. When grid clustering was increased to $AR_{\max} = 10,000$, the LSBMR converged consistently faster than other acceleration schemes (Fig. 8). It is interesting to notice that TSS did not this time have any effect when applied to the basic Runge-Kutta scheme, while achieving impressive acceleration (Fig. 8) when applied together with the LSBMR method. To test the acceleration methods for situations with flow separation that stretches through the exit plane^{22 23} we computed laminar flow in a U-shaped channel with computational grid symmetrically clustered towards the channel walls. With $AR_{\max} = 200$,

LSBMR method offered the fastest convergence (Fig. 9) in this case, while IRS required more time than the RK scheme although coefficients of the IRS were chosen for maximum performance¹.

Performance of the SBMR method was next evaluated against the DMR method and the RK scheme for inviscid compressible flows. Convergence rates of existing algorithms are very slow when integrating compressible flow equations at low Mach numbers¹⁵ because of the excessive stiffness of the system. We used second order accurate central differencing for spatial discretizations at interior grid points, while a second order accurate one-sided differencing was used at the boundary points. A combination of second and fourth order artificial dissipation¹ was used in a fully conservative form to eliminate even-odd decoupling. A flow tangency boundary condition was employed on the solid boundaries. The test geometry was a straight channel with the computational grid composed of 64×32 cells clustered towards the bottom wall having a circular arc bump⁴. Grid cells in the main stream direction were clustered in the constant height sections of the channel towards the leading edge and the trailing edge of the arc, while a uniform grid spacing was used in the arc section. In all test cases both SBMR and DMR methods were applied after every 30 iterations. The DMR method combined two consecutive iteration steps, while the SBMR method used the results of $n = 5$ iteration levels apart except for the case of inlet Mach number $M_\infty = 0.675$ where the results of $n = 10$ levels apart were used. The convergence histories for $M_\infty = 0.05$ (Fig. 10) are equally impressive for both SBMR and DMR methods. They reduced the number of iterations and the computing time by 84% for this stiff system. The actual CPU time overhead per iteration for the SBMR method was found to be 8.8%, while it was 2.4% for the DMR method for the test cases done here. The transonic ($M_\infty = 0.675$) test case required five times larger artificial dissipation to stabilize the solution with a shock wave²² and convergence of the basic RK iteration scheme became much faster than for the low subsonic case (Fig. 11) and no convergence acceleration was obtained by either the SBMR method or the DMR method. Since the present formulation of the SBMR method does not account for the artificial dissipation terms, the convergence of the basic RK scheme with smaller amount of artificial dissipation for the transonic flow becomes slower^{29,30}, while the SBMR method still offers convergence acceleration (Fig. 12).

B. Three-Dimensional Flow

To compare performances of the SBMR, PSBMR and APSBMR methods, a fully developed laminar incompressible flow was computed through a straight duct with unit square cross section having length of ten units. The grid was clustered symmetrically toward the solid walls and a uniform grid spacing was used in the axial x-direction. Chorin's compressibility coefficient $\beta = 5$ was used in this case. At the inlet boundary, a fully developed three-dimensional laminar velocity profile was given and pressure was computed from characteristic boundary conditions, while a non-reflecting boundary condition was used at the exit boundary²⁵. The SBMR method was applied after every 50 iterations utilizing the results $n = 10$ iteration levels apart. The convergence histories for $AR_{\max} = 127$ (Fig. 13) and $AR_{\max} = 318$ (Fig. 14) demonstrate that PSBMR and especially APSBMR method are capable of computing time reductions close to 50% although optimum acceleration parameters for SBMR rapidly stabilize (Fig. 15), while these parameters for APSBMR in clustered and coarse grid regions continue changing (Fig. 16) throughout the iterative process.

V. Conclusions and Recommendations

The sensitivity-based minimal residual methods and its variations are applicable to a general system of time-dependent second order partial differential equations and systems of such equations. The performance of the SBMR and the LSBMR methods depends on how frequently these methods are applied during the basic iteration process and on the number of iterations performed with the basic iterative algorithm that are involved in the evaluation of the change of the solution vector. In the case of two-dimensional incompressible viscous flows without severe pressure gradient, the SBMR and LSBMR methods significantly accelerate the convergence of iterative procedure on clustered grids with the LSBMR method becoming more efficient as grids are becoming highly clustered. For a two-dimensional incompressible laminar flow, the SBMR and

the LSBMR methods offer fast convergence regardless of the value of the artificial compressibility, β , when it is of order one. Also, the SBMR and the LSBMR methods tested for a two-dimensional incompressible laminar flow maintain the fast convergence for highly non-orthogonal grids and for flows with closed and open flow separation. The TSS method, used together with the basic iteration scheme and the LSBMR method, greatly enhances the convergence rate for solving a two-dimensional incompressible laminar and turbulent flows through a straight channel with highly clustered grids. The SBMR method is capable of accelerating the convergence of inviscid, low Mach number, compressible flows where the system is very stiff. However, the SBMR method can not offer any convergence acceleration for a transonic flow where there is a large amount of artificial dissipation. The APSBMR method, a three-dimensional analogy of the LSBMR method, successfully reduces the computational effort for solving a three-dimensional, laminar flow through a straight duct without flow separation. The current formulation of the sensitivity-based methods do not include the effects of the artificial dissipation due to the complexity of such formulations. The general formulation of the new acceleration methods is applicable to any iteration scheme (explicit or implicit) as the basic iteration algorithm. Hence, it should be possible to apply the new acceleration methods in conjunction with the other iteration algorithms and with other acceleration methods to explore the possibilities for a cumulative acceleration effect.

Acknowledgments

This work was supported by the grant NAS 8-38861 from NASA MSFC, Huntsville, AL monitored by Dr. Paul McConaughy and by the Penn State NASA Propulsion Engineering Research Center, contract NAGW-1356, Supplement 8. Computing time was provided on the NASA Ames NAS Cray-90 facility by Dr. Louis Povinelli of NASA Lewis Research Center. The final text proofreading was performed by Mr. Thomas J. Martin and Ms. Amy Myers.

References

- ¹Jameson, A., Schmidt, W. and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, CA, June 1981.
- ²Pulliam, T. H., "Euler and Thin Layer Navier-Stokes Codes: ARC2D, ARC3D," Notes for Computational Fluid Dynamics User's Workshop, The University of Tennessee Space Institute, Tullahoma, TN, March 1984.
- ³Jameson, A. and Baker, T. J., "Solution of the Euler Equations for Complex Configurations," AIAA Paper 83-1929, Danvers, MA, July 1983.
- ⁴Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, Vol. 20, No. 11, 1982, pp. 1565-1571.
- ⁵Martinelli, L., Jameson, A. and Grasso, F., "A Multigrid Method for the Navier-Stokes Equations," AIAA Paper 86-0208, Reno, NV, January 1986.
- ⁶Caughey, D. A., "A Diagonal Implicit Multigrid Algorithm for the Euler Equations," AIAA Paper 87-0354, Reno, NV, January 1987.
- ⁷Chima, R. V. and Turkel, E. and Schaffer, S., "Comparison of Three Explicit Multigrid Methods for the Euler and Navier-Stokes Equations," AIAA Paper 87-0602, Reno, NV, January 1987.
- ⁸Dannenhoffer, J. F. and Giles, M. B., "Accelerated Convergence of Euler Solutions Using Time Inclining," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1457-1463.
- ⁹Turkel, E., "Review of Preconditioning Methods for Fluid Dynamics," ICASE Report No. 92-47, NASA Langley Research Center, Hampton, VA, September 1992.
- ¹⁰Choi, Y. and Merkle, C. L., "Time-Derivative Preconditioning for Viscous Flow," AIAA Paper 91-1652, Honolulu, HI, June 1991.

- ¹¹Kennon, S. R. and Dulikravich, G. S., "Optimum Acceleration Factors for Iterative Solution of Linear and Nonlinear Differential Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 47, 1984, pp. 357-367.
- ¹²Hafez, M., Parlette, E. and Salas, M. D., "Convergence Acceleration of Iterative Solutions of Euler Equations for Transonic Flow Computations," AIAA Paper 85-1641, Cincinnati, OH, July 1985.
- ¹³Huang, C. Y., Kennon, S. R. and Dulikravich, G. S., "Generalized Non-Linear General Minimal Residual (GNLMR) Method for Iterative Algorithms", *Journal of Computational and Applied Mathematics*, Vol. 16, Nov. 1986, pp. 215-232.
- ¹⁴Huang, C. Y. and Dulikravich, G. S., "Fast Iterative Algorithms Based on Optimized Explicit Time-Stepping," *Computer Methods in Applied Mechanics and Engineering*, Vol. 63, August 1987, pp. 15-36.
- ¹⁵Lee, S. and Dulikravich, G. S., "Acceleration of Iterative Algorithms for Euler Equations of Gasdynamics," *AIAA Journal*, Vol. 28, No. 5, 1990, pp. 939-942.
- ¹⁶Lee, S. and Dulikravich, G. S., "Distributed Minimal Residual (DMR) Method for Acceleration of Iterative Algorithms," *Computer Methods in Applied Mechanics and Engineering*, Vol. 86, 1991, pp. 245-262.
- ¹⁷Lee, S. and Dulikravich, G. S., "Accelerated Computation of Viscous Flow with Heat Transfer," *Numerical Heat Transfer: Fundamentals, Part B*, Vol. 19, June 1991, pp. 223-241.
- ¹⁸Saad, Y. and Schultz, M., "Conjugate Gradient-Like Algorithms for Solving Non-symmetric Linear Systems," *Mathematics of Computation*, Vol. 44, No. 170, 1985, pp. 417-424.
- ¹⁹Choi, K. Y. and Dulikravich, G. S., "Reliability Enhancement of Navier-Stokes Codes Through Convergence Enhancement," *Proceedings of 5th Annual Symposium of the Penn State - NASA Propulsion Eng. Research Center*, NASA Propulsion Engineering Research Center, University Park, PA, September 1993

- ²⁰Choi, K. Y. and Dulikravich, G. S., "Global Sensitivity-Based Minimal Residual (GSBMR) Method for Acceleration of Iterative Convergence Rates," *Proceedings of the 14th International Conference on Numerical Fluid Dynamics*, Bangalore, India, July 1994.
- ²¹Choi, K. Y. and Dulikravich, G. S., "Convergence Rate Enhancement of Navier-Stokes Codes on Clustered Grids", *Proceedings of 6th Annual Symposium of the Penn State - NASA Propulsion Eng. Research Center*, NASA Lewis Research Center, Cleveland, OH, September 1994.
- ²²Choi, K.-Y., "Sensitivity-Based Methods for Convergence Acceleration of Iterative Algorithms," Ph. D. Thesis, The Department of Aerospace Engineering, The Pennsylvania State University, December 1994.
- ²³Choi, K. Y. and Dulikravich, G. S., "Sensitivity-Based Methods for Convergence Acceleration of Iterative Algorithms" in press: *Computer Methods in Applied Mechanics and Engineering*.
- ²⁴Thompson, K. W., "Time-dependent Boundary Conditions for Hyperbolic Systems, II," *Journal of Computational Physics*, Vol. 89, 1990, pp. 439-461.
- ²⁵Dulikravich, G.S., Ahuja, V. and Lee, S., "Modeling Three-dimensional Solidification with Magnetic Fields and Reduced Gravity," *International Journal of Heat and Mass Transfer*, Vol. 37, No. 5, 1994, pp. 837-853.
- ²⁶Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, 1967, pp. 12-26.
- ²⁷MacCormack, R.W. and Baldwin, B.S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interaction," AIAA Paper 75-1, Jan. 1975.
- ²⁸Baldwin, B.S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Huntsville, AL, January 1978.
- ²⁹Dulikravich, G. S., Dorney, D. J. and Lee, S., "Numerical Versus Physical Dissipation in the Solution of Compressible Navier-Stokes Equations," AIAA Paper 89-0550, Reno, NV, January 1989.
- ³⁰Dulikravich, G. S. and Dorney, D. J., "Artificial Dissipation Sensors for Computational Gasdynamics," AIAA Paper 89-0643, Reno, NV, January 1989.

LIST OF FIGURE CAPTIONS

- Fig. 1 Convergence histories for a straight channel incompressible flow ($Re = 1600$, $L/H = 5$, 60×60 grid cells, $AR_{\max} = 100$, $\beta = 5$, $CFL = 2.8$, $\sigma = 0.4$).
- Fig. 2 Evolution of optimum acceleration parameters for LSBMR method applied to a straight channel laminar incompressible flow: a) in a clustered grid region, b) in a coarse grid region.
- Fig. 3 Grid non-orthogonality effect on RK, SBMR and LSBMR convergence: a) a non-orthogonal clustered grid, b) sensitivity to non-orthogonality angle.
- Fig. 4 Effect of frequency of application of: a) SBMR, and b) LSBMR on convergence.
- Fig. 5 Effect of the number of iterations apart, n , appearing in the SBMR and LSBMR formulations on the convergence of: a) SBMR, b) LSBMR.
- Fig. 6 Convergence histories for a straight channel incompressible flow ($Re = 1600$, $L/H = 5$, 60×60 grid cells, $AR_{\max} = 10,000$, $\beta = 5$, $CFL = 2.8$, $\sigma = 0.4$).
- Fig. 7 Convergence histories for a straight channel incompressible turbulent flow ($Re = 1.6 \times 10^6$, $L/H = 10$, 60×120 grid cells, $AR_{\max} = 1,000$, $\beta = 5$, $CFL = 2.0$, $\sigma = 0.4$).
- Fig. 8 Convergence histories for a straight channel incompressible turbulent flow ($Re = 1.6 \times 10^6$, $L/H = 10$, 60×120 grid cells, $AR_{\max} = 10,000$, $\beta = 5$, $CFL = 2.0$, $\sigma = 0.4$).
- Fig. 9 Convergence histories for a U-shaped channel incompressible flow ($Re = 200$, 129×80 grid cells, $AR_{\max} = 200$, $\beta = 10$, $CFL = 2.8$, $\sigma = 0.4$).
- Fig. 10 Convergence histories for the compressible flow Euler equations ($M_{\infty} = 0.05$).
- Fig. 11 Convergence histories for the compressible flow Euler equations ($M_{\infty} = 0.675$).

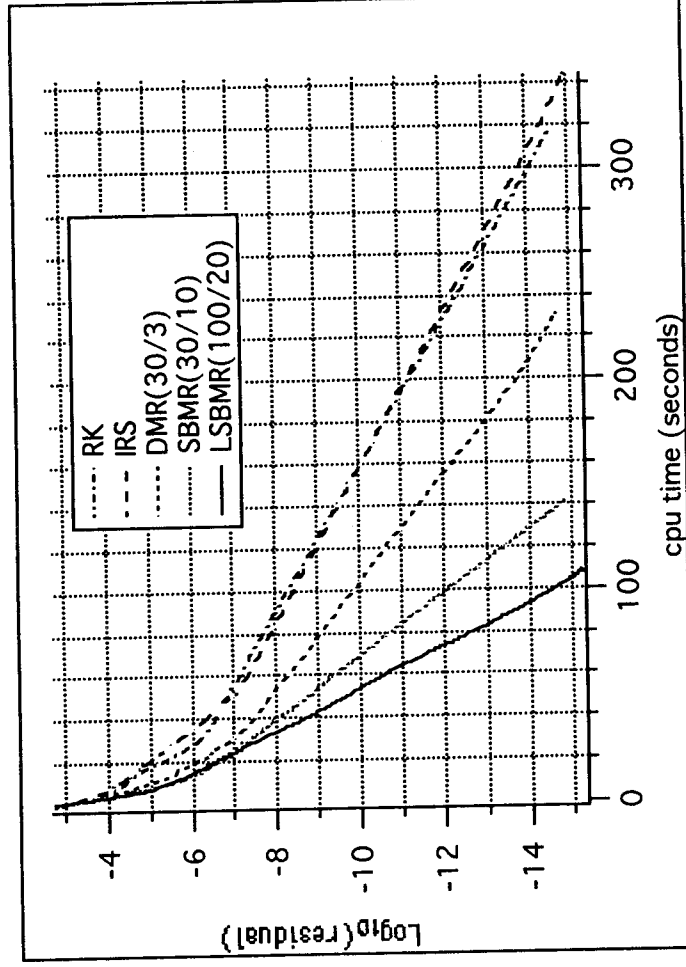
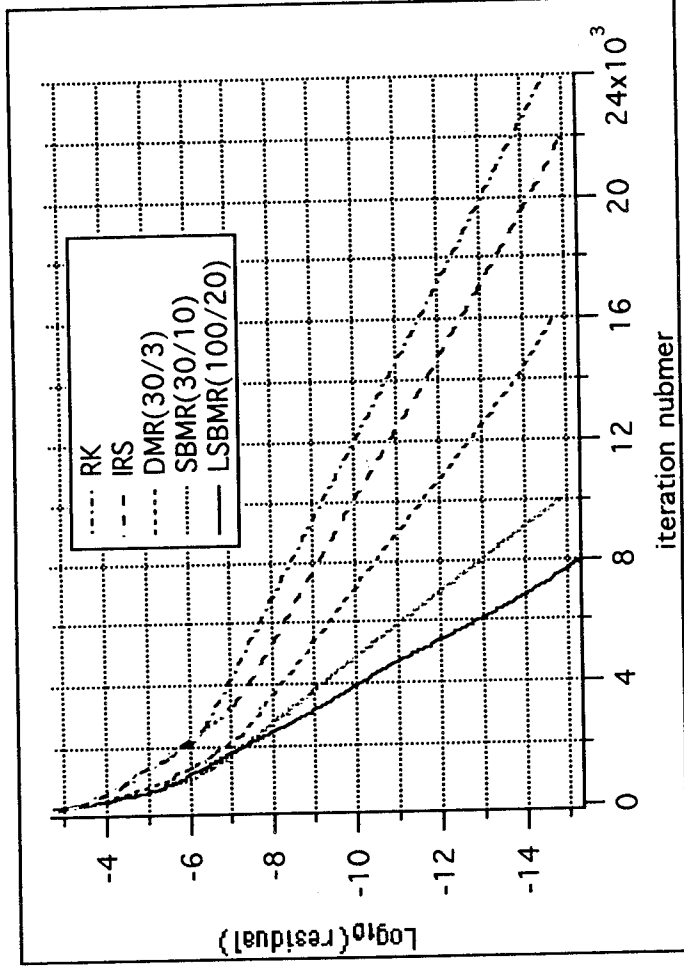
Fig. 12 Convergence histories for the compressible flow Euler equations ($M_\infty = 0.675$) with smaller artificial dissipation.

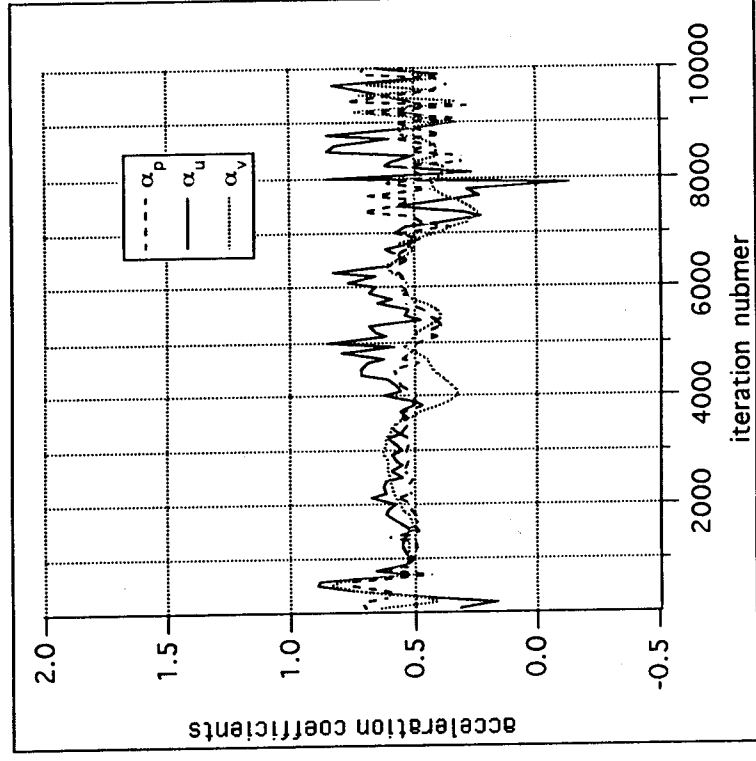
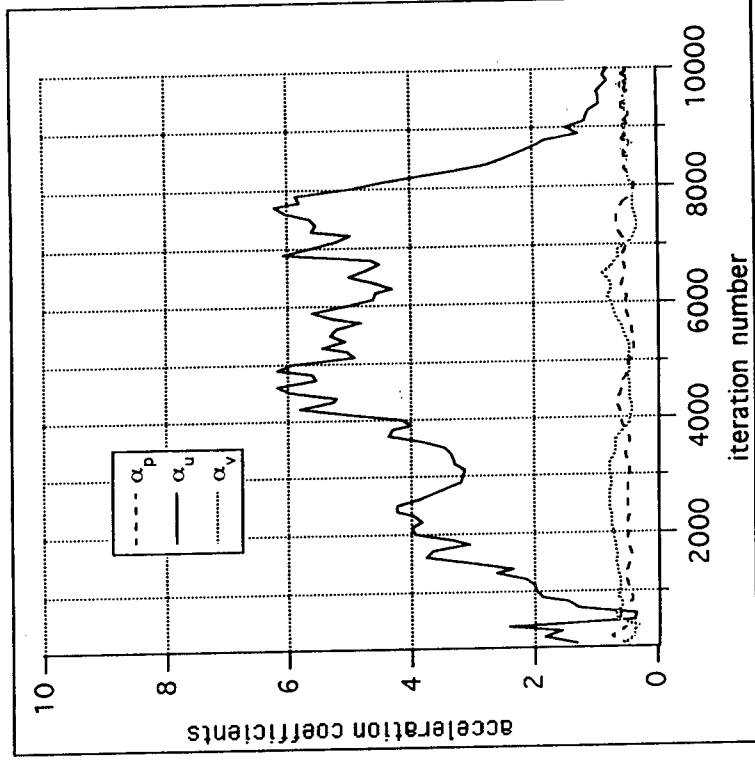
Fig. 13 Convergence histories for a straight square duct incompressible flow ($Re = 2000$, $24 \times 20 \times 20$ grid cells, $AR_{\max} = 127$).

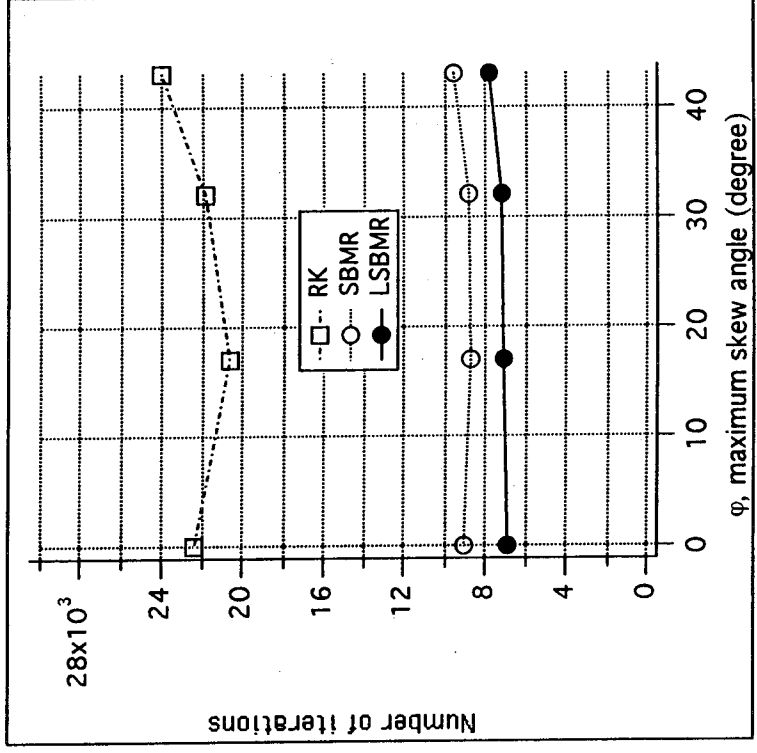
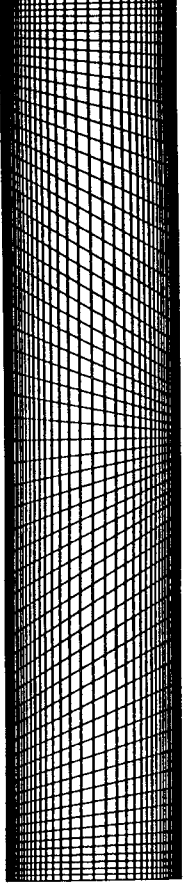
Fig. 14 Convergence histories for a straight square duct incompressible flow ($Re = 2000$, $24 \times 20 \times 20$ grid cells, $AR_{\max} = 318$).

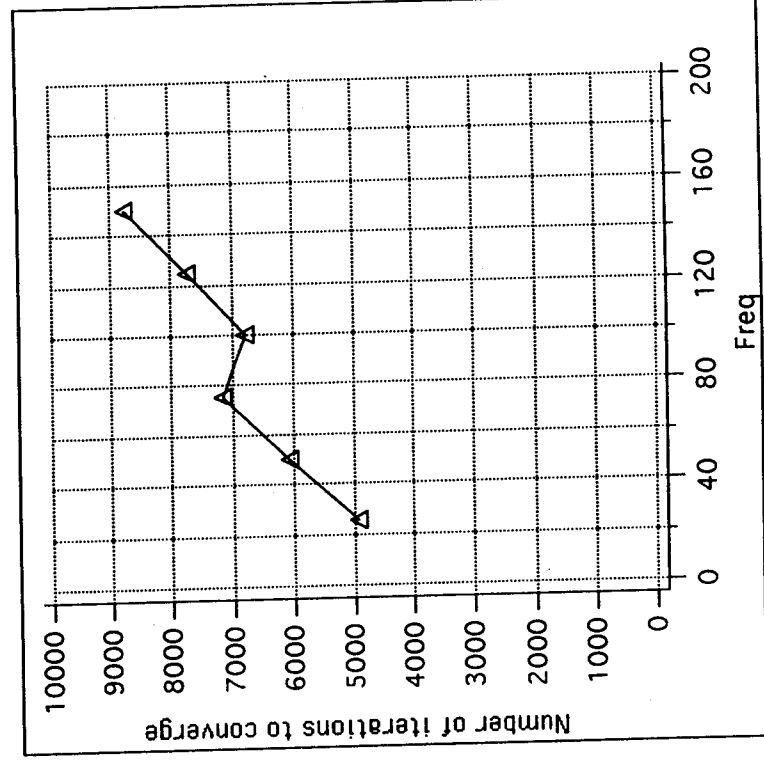
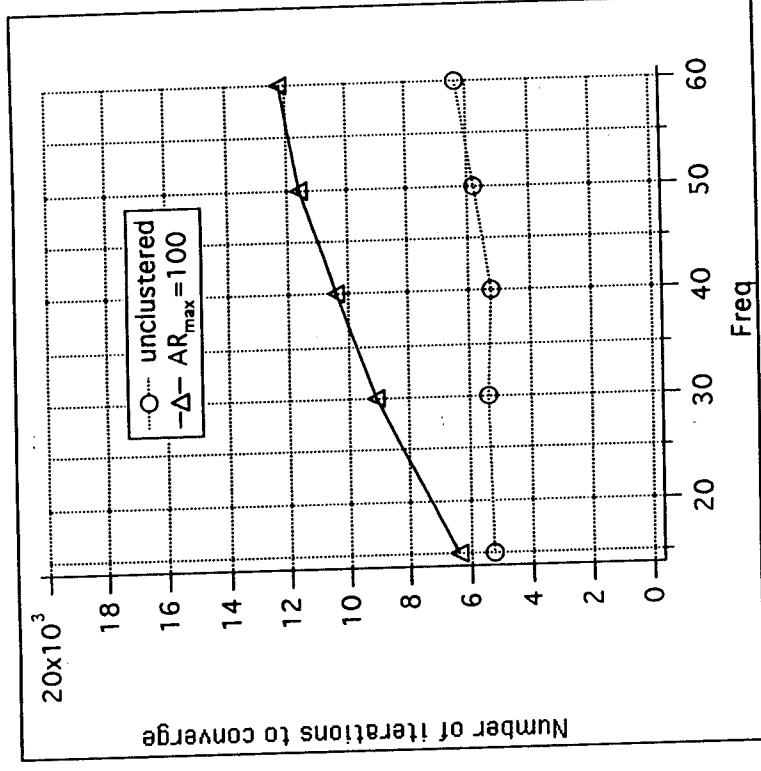
Fig. 15 Evolution of optimum acceleration parameters for SBMR method applied to a laminar incompressible flow in a straight square duct ($\beta = 5$, $AR_{\max} = 318$, $Freq = 50$, $n = 10$).

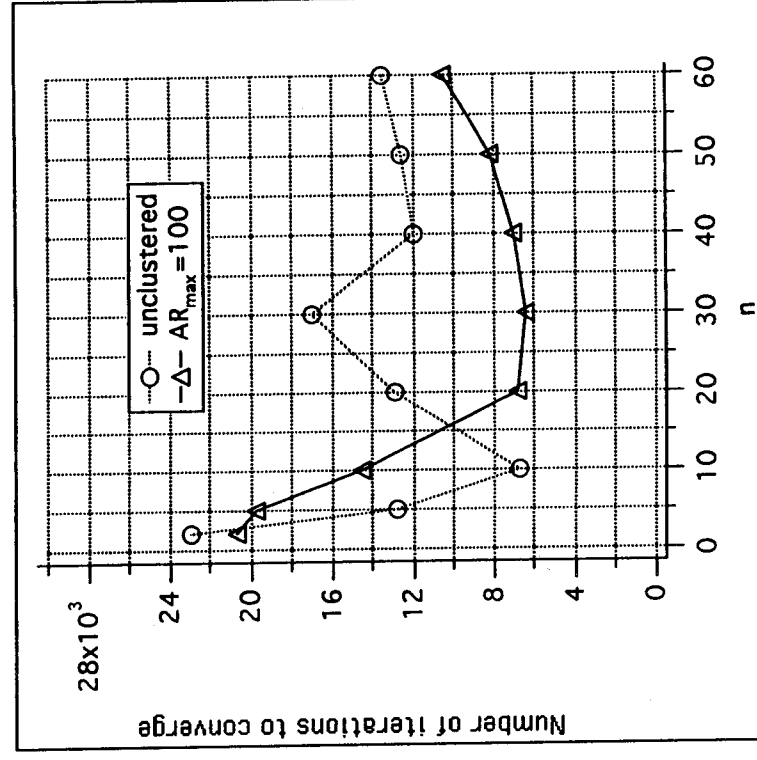
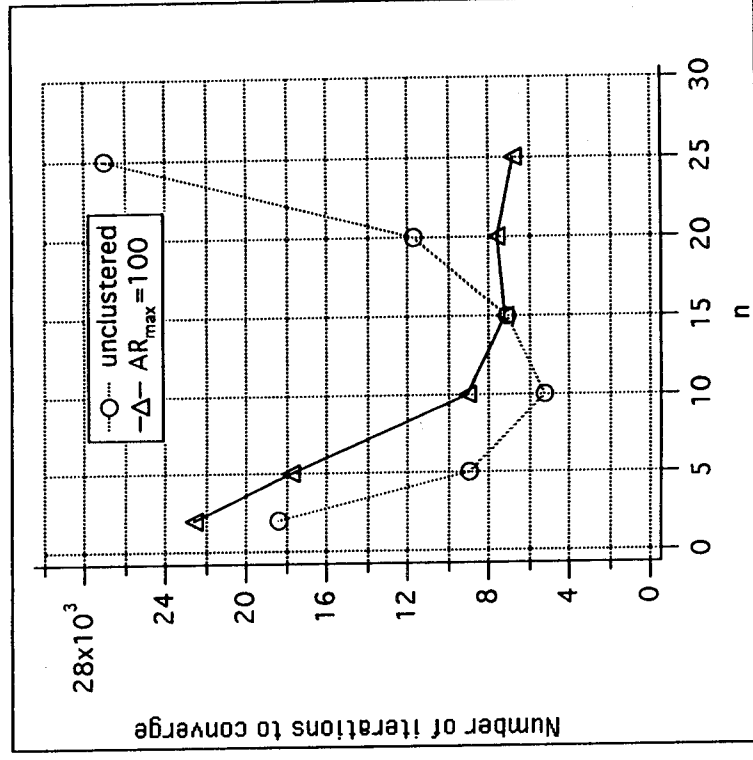
Fig. 16 Evolution of optimum acceleration parameters for APBMR method applied to a laminar incompressible flow in a straight square duct ($\beta = 5$, $AR_{\max} = 318$, $Freq = 100$, $n = 10$).

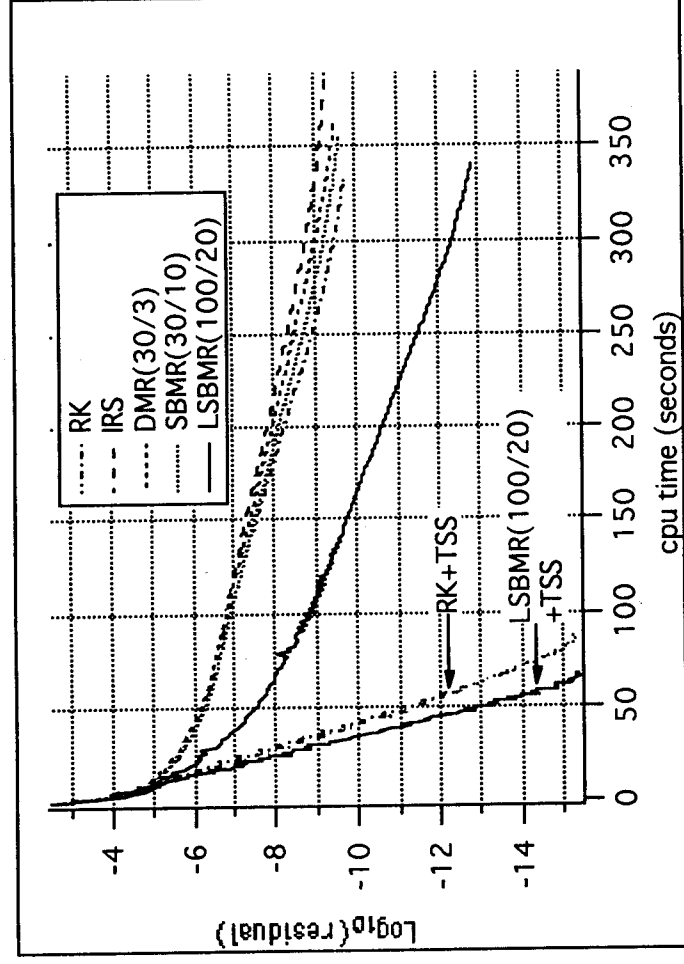
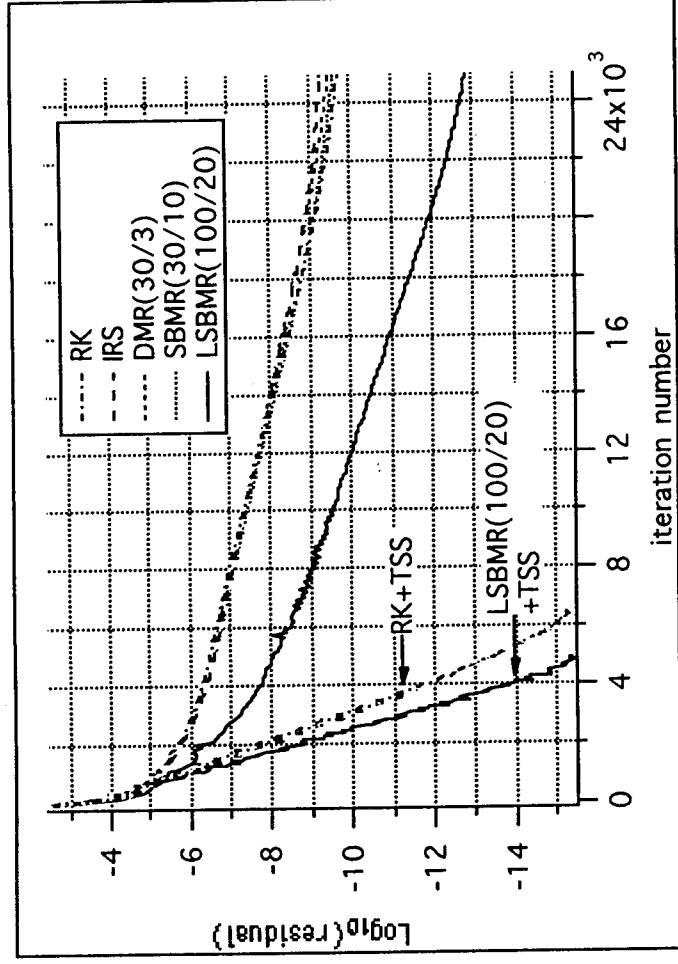


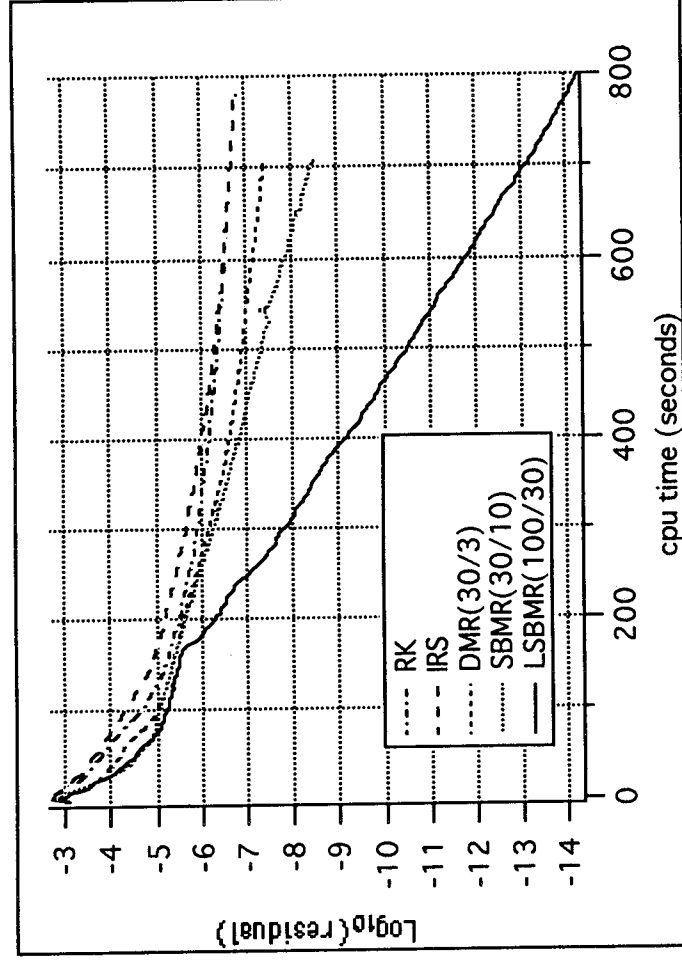
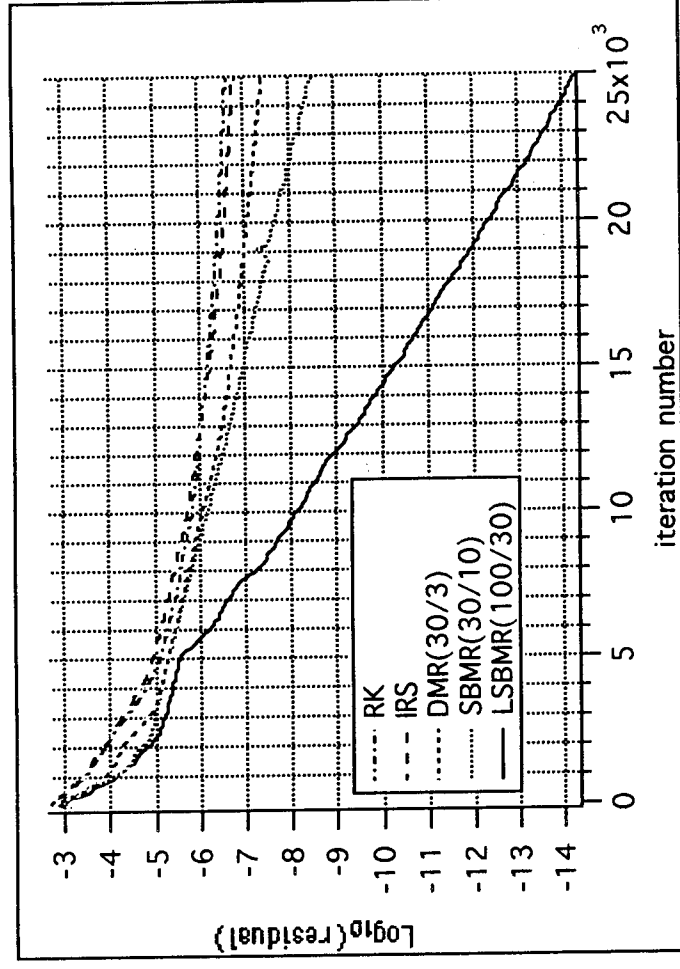


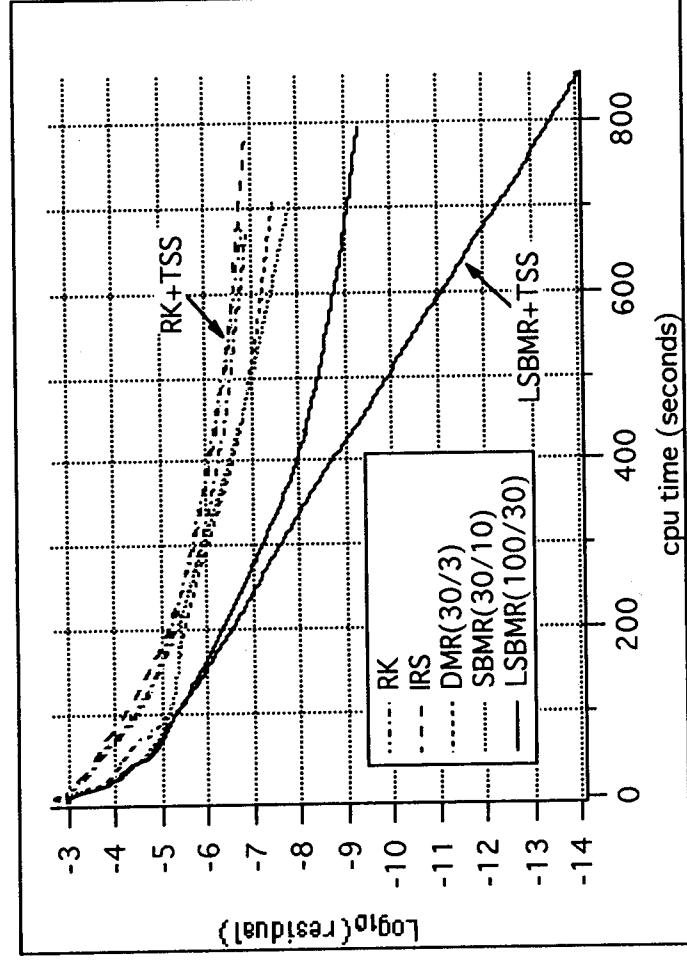
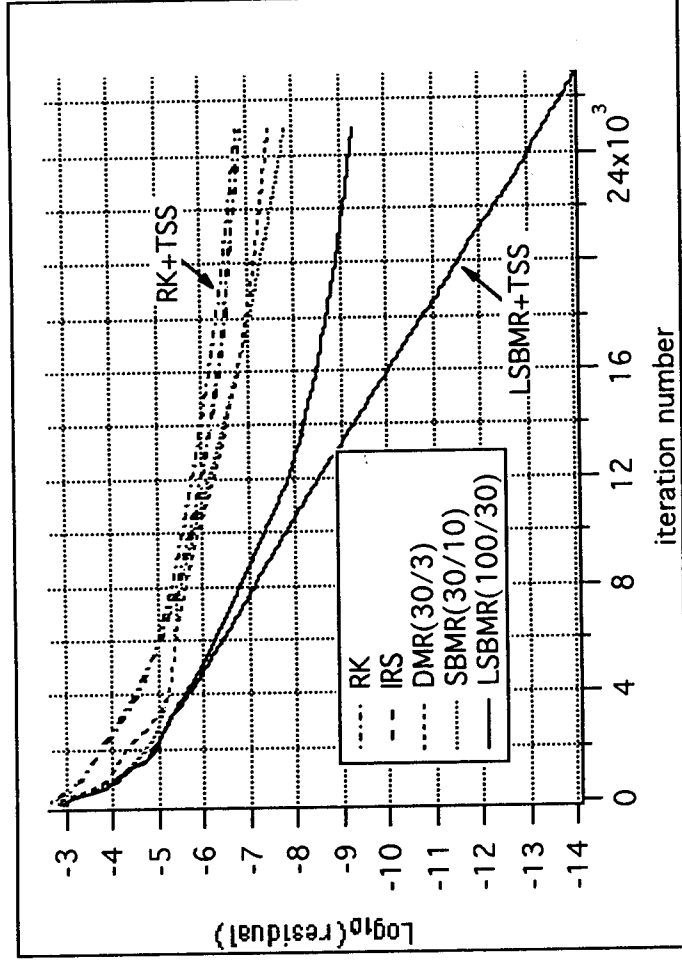


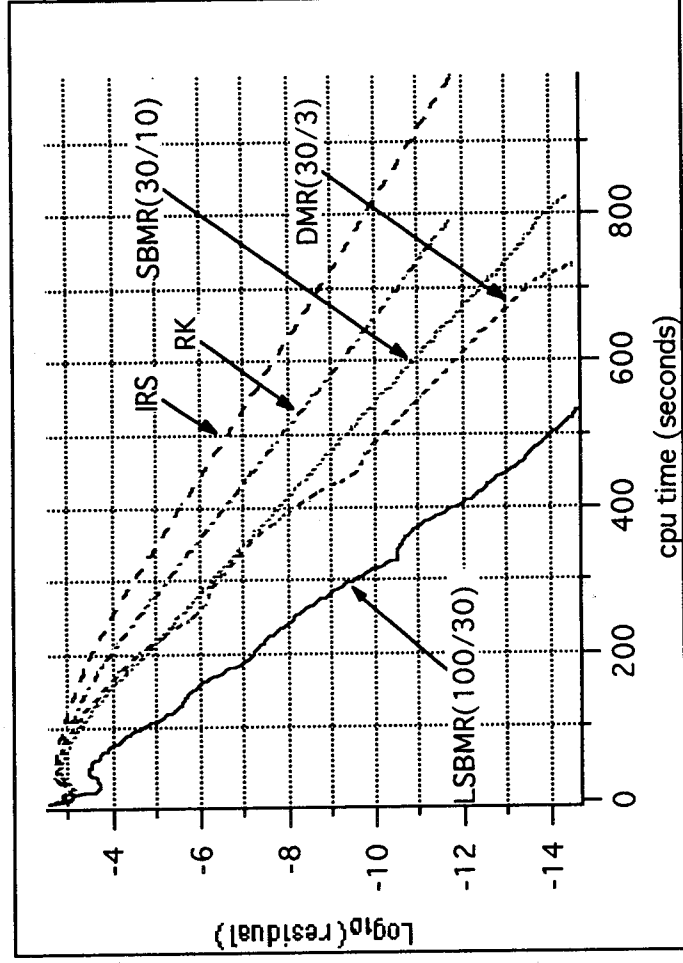
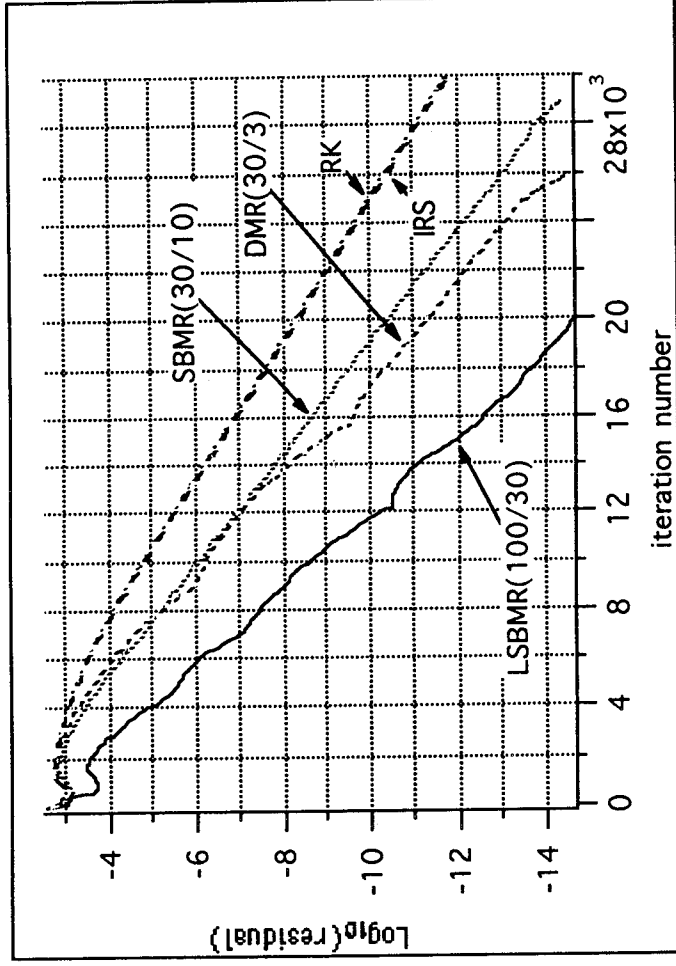


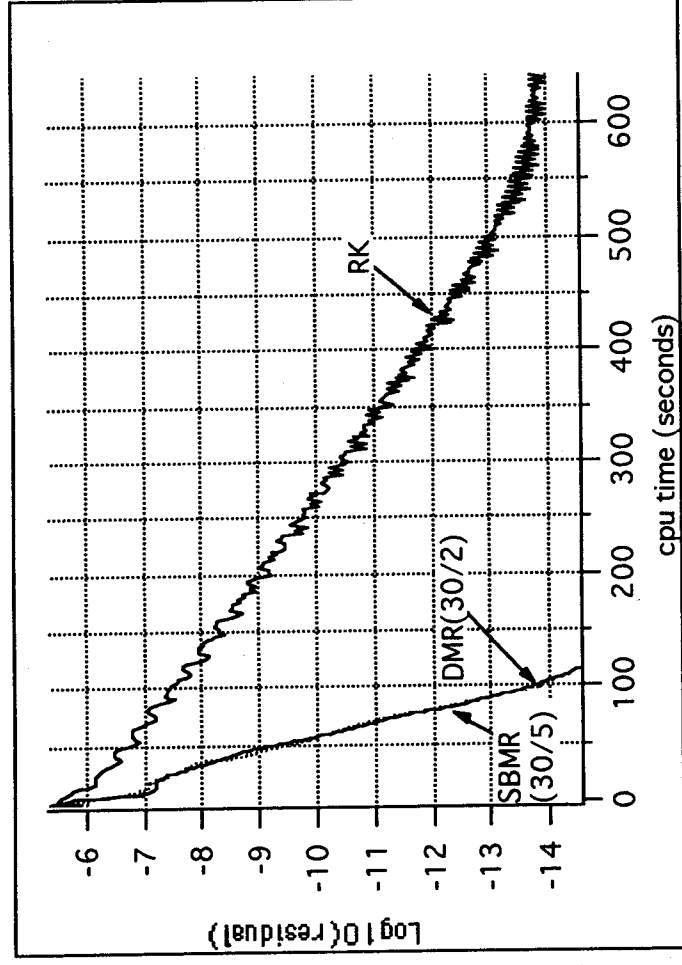
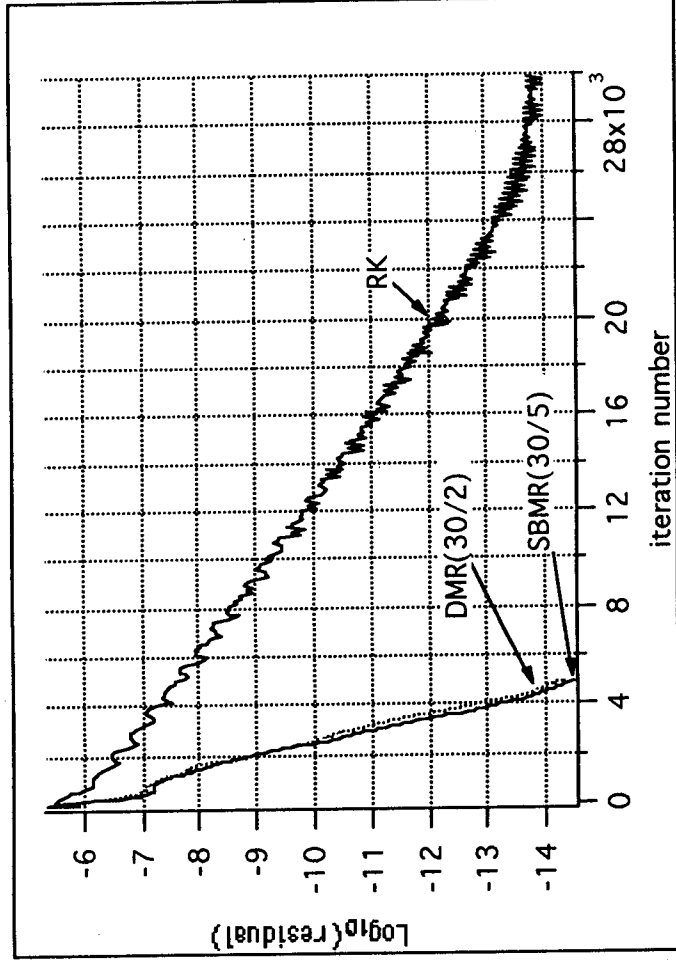


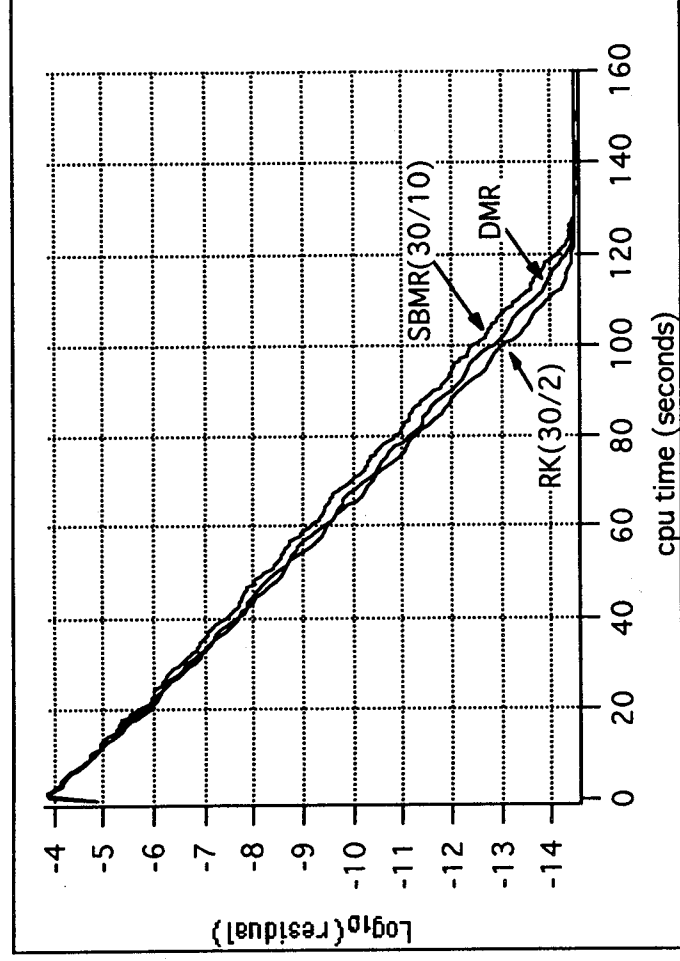
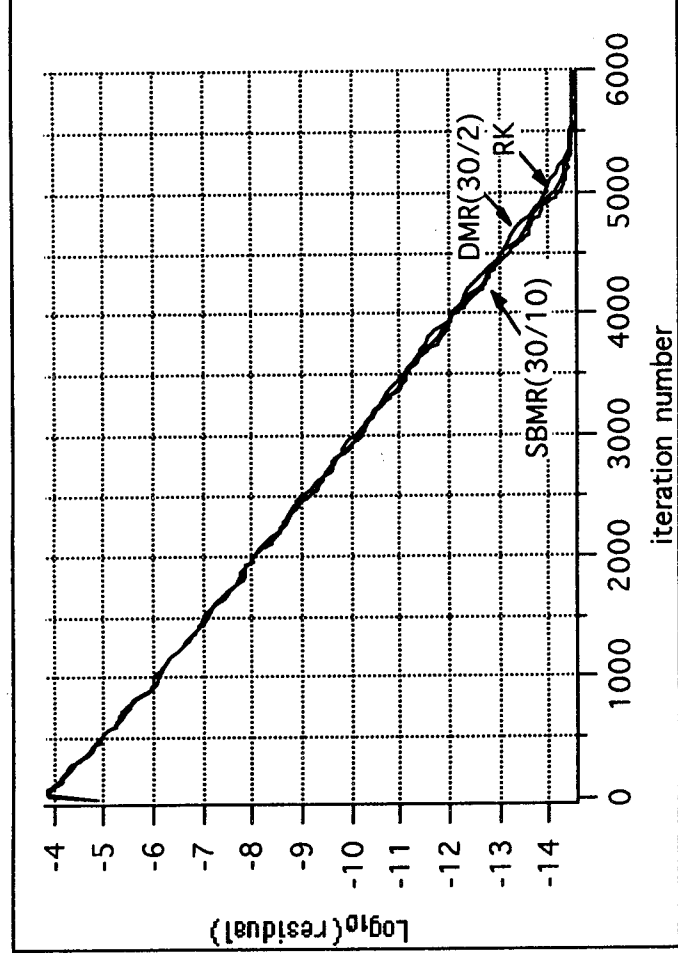


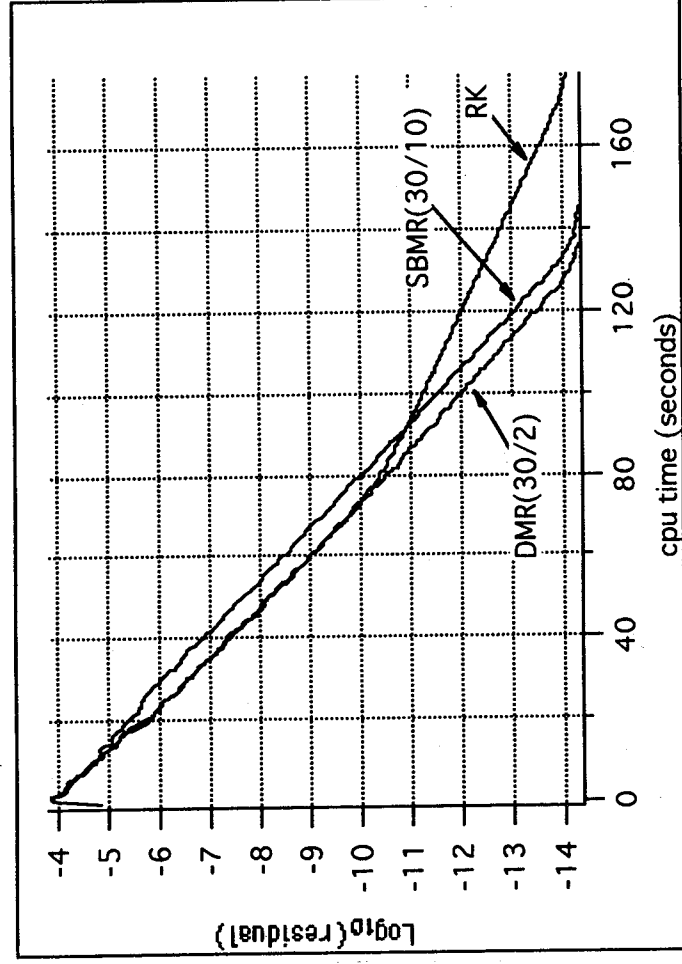
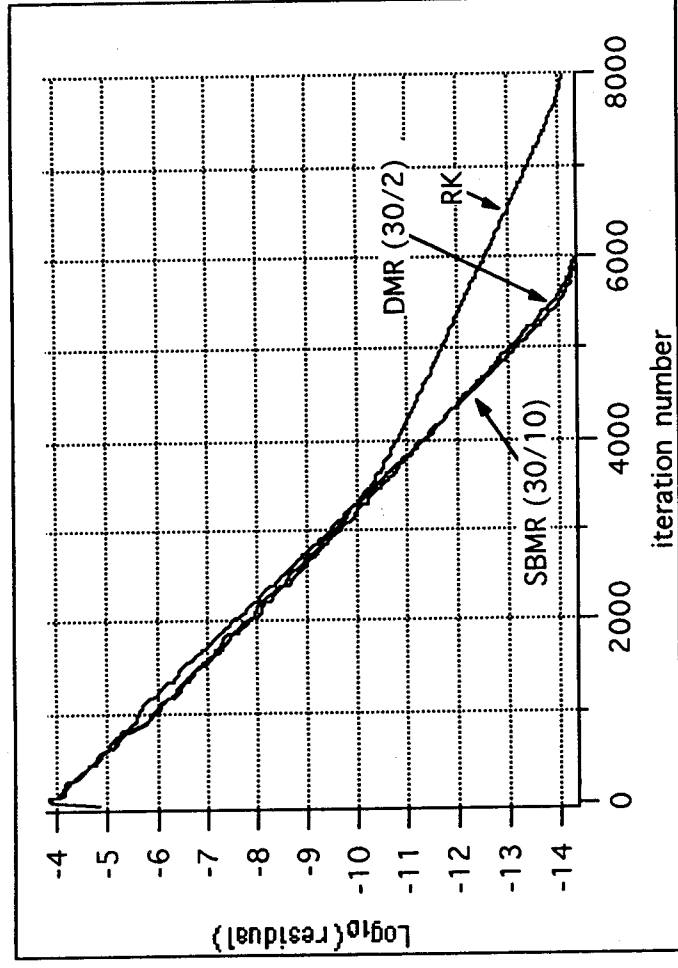


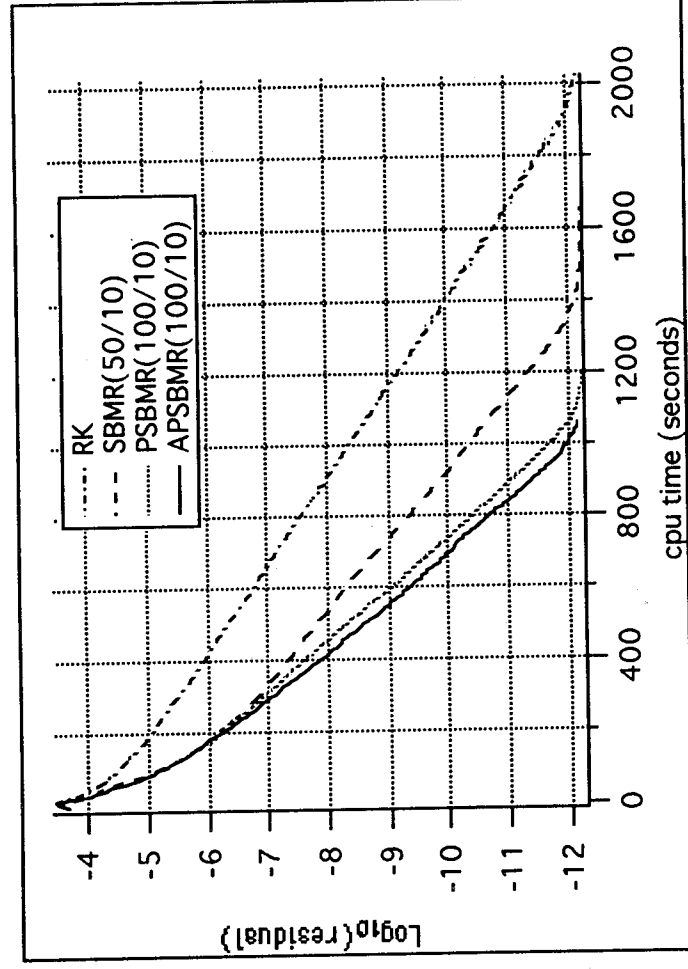
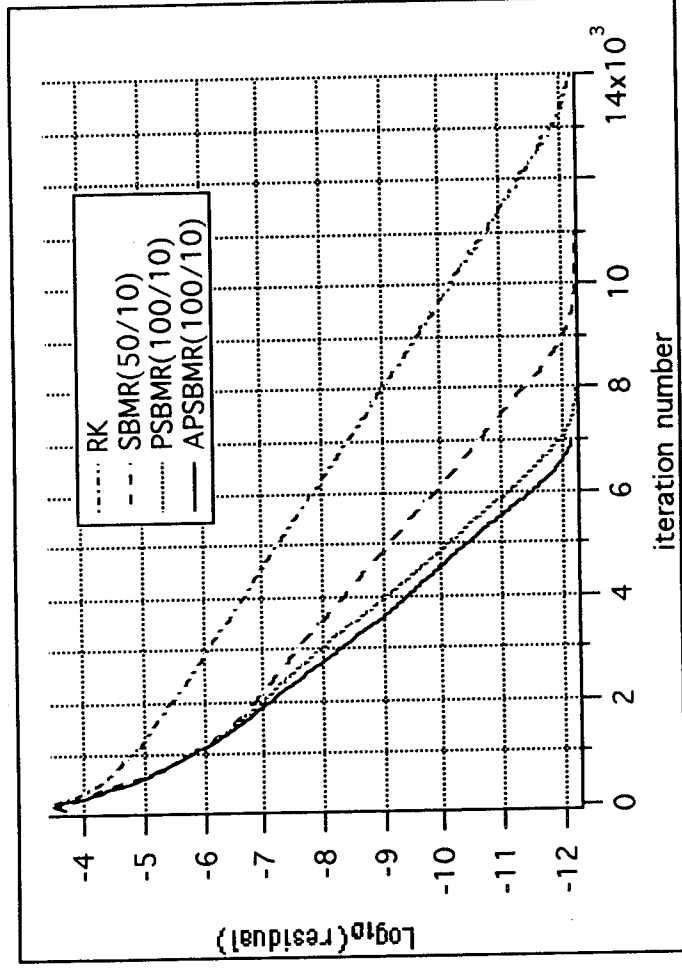


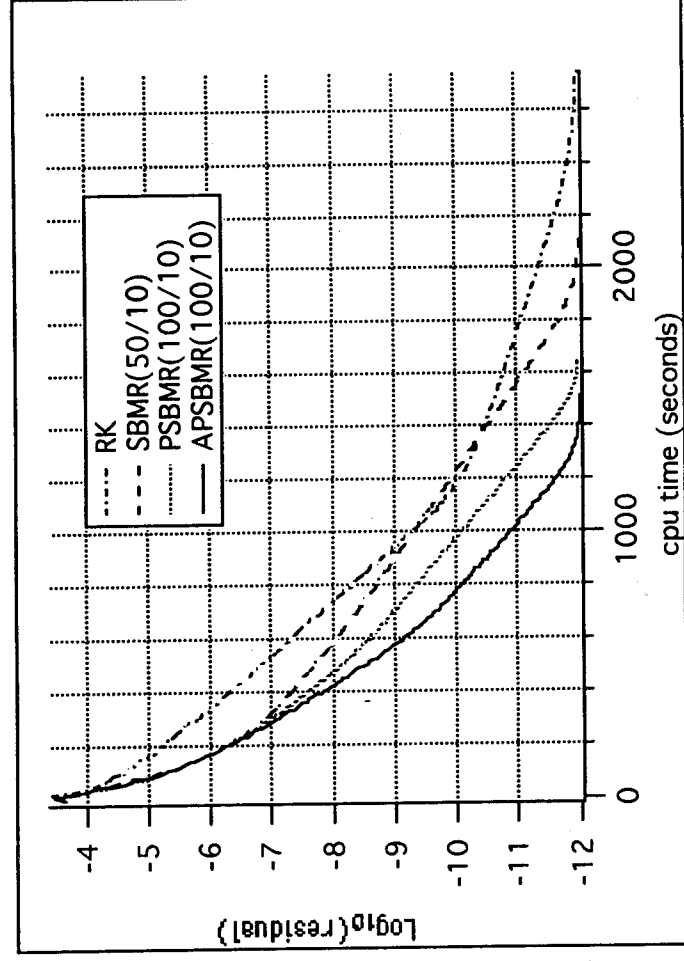
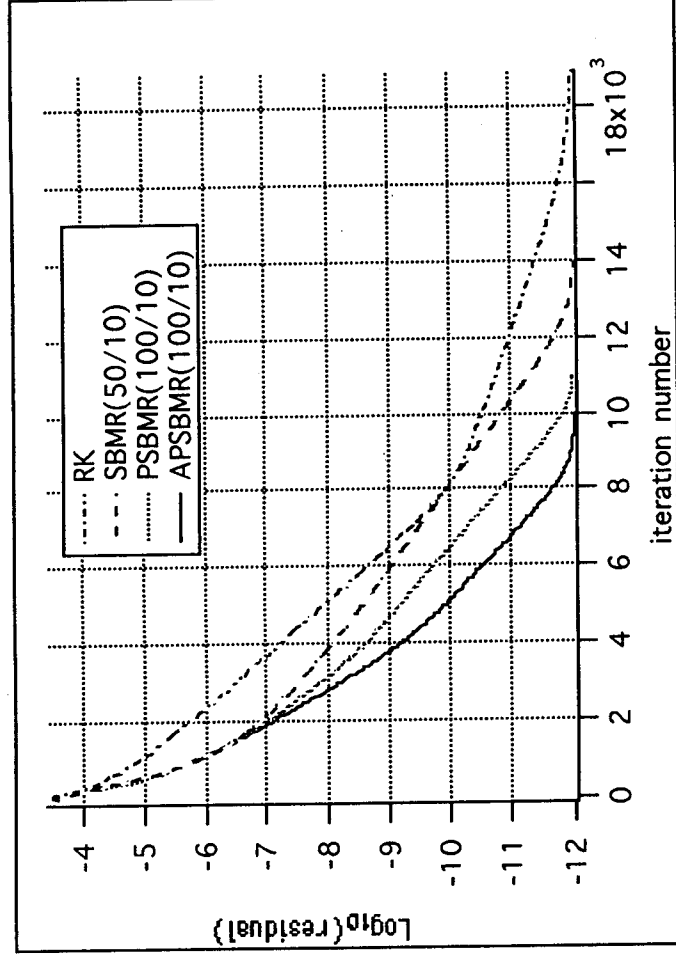


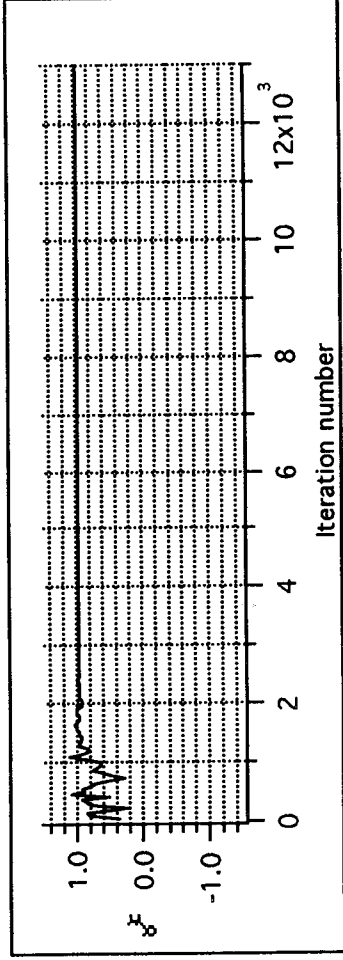
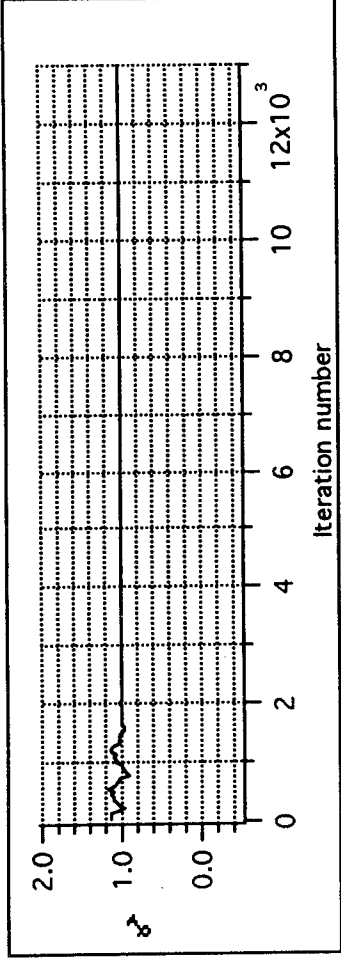
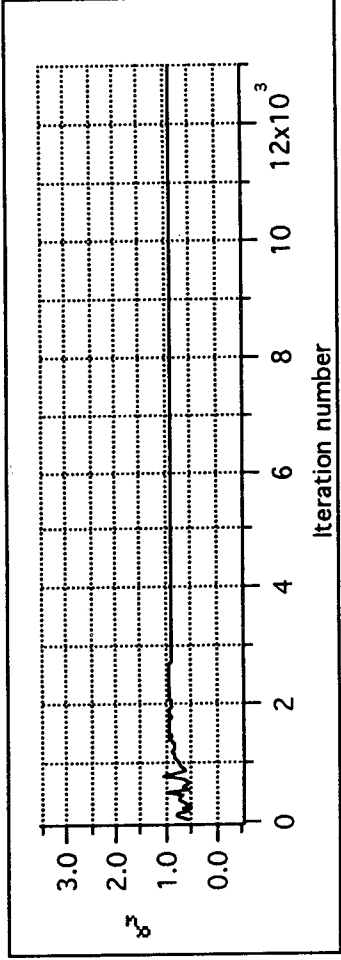
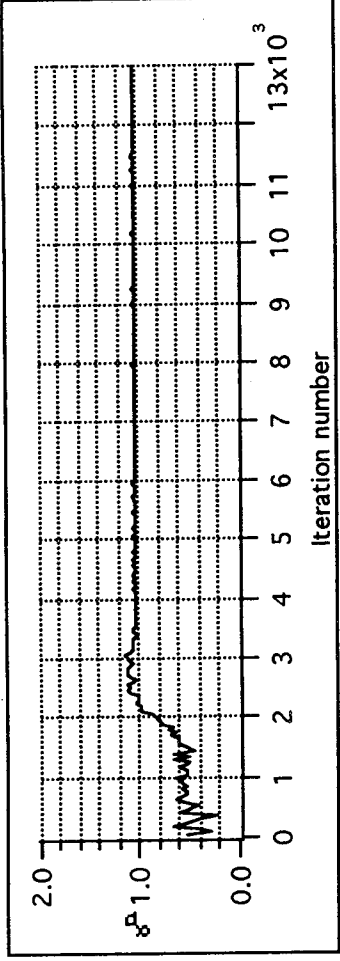


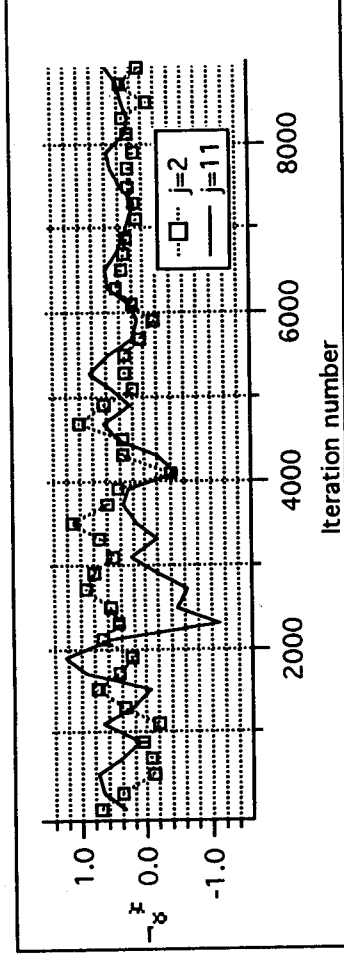
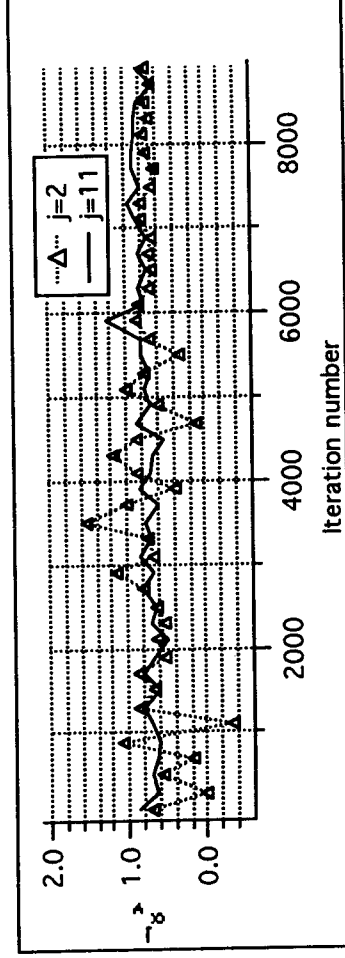
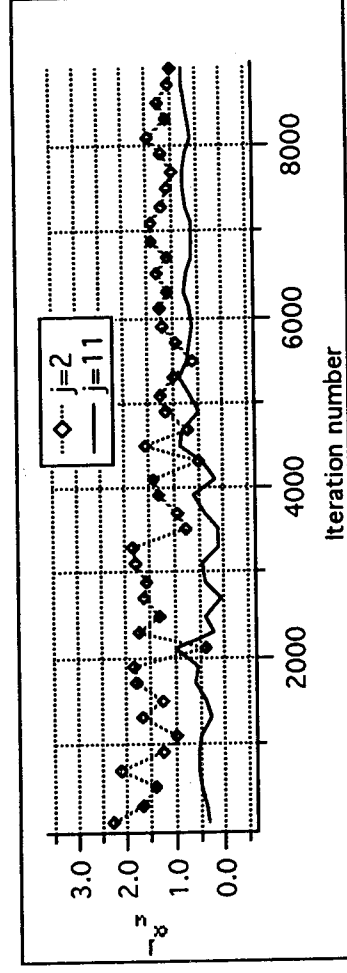
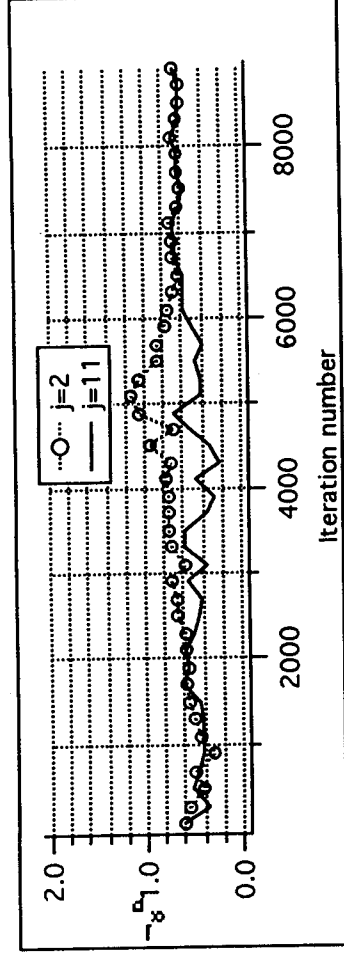












(0) (2)
CN22
TURNER J/PUBLICATION
MARSHALL SPACE FLIGHT CENTER
HUNTSVILLE AL.

DELETIONS OR CHANGES 544-4494
RETURN ADDRESS CN22D 000002444